AFRL-VA-WP-TM-2006-3187

INTEGRATED DESIGN AND ANALYSIS TOOLS FOR SOFTWARE-BASED CONTROL SYSTEMS

Dr. Shankar Sastry
Dr. Thomas Henzinger
Dr. Edward Lee

University of California at Berkeley
391 Corry Hall
Berkeley, CA 94720

JULY 2005

Final Report for 08 August 1998 – 31 December 2004

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE AND SIGNATURE PAGE

AFRL-VA-WP-TM-2006-3187 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//Signature//

RAYMOND A. BORTNER
Senior Electronic Engineer

//Signature//

MICHAEL P. CAMDEN, Chief
Control Systems Development and
Applications Branch
Control Sciences Division

//Signature//

JEFFREY C. TROMP
Senior Technical Advisor
Control Sciences Division
Air Vehicles Directorate

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS**.

1. REPORT DATE ***(DD-MM-YY)*** July 2005	**2. REPORT TYPE** Final	**3. DATES COVERED** ***(From - To)*** 08/08/1998– 12/31/2004

4. TITLE AND SUBTITLE

INTEGRATED DESIGN AND ANALYSIS TOOLS FOR SOFTWARE-BASED CONTROL SYSTEMS

5a. CONTRACT NUMBER
F33615-98-C-3614

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62301E

6. AUTHOR(S)

Dr. Shankar Sastry
Dr. Thomas Henzinger
Dr. Edward Lee

5d. PROJECT NUMBER
A04H

5e. TASK NUMBER

5f. WORK UNIT NUMBER
0A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of California at Berkeley
391 Corry Hall
Berkeley, CA 94720

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

Defense Advanced Research Projects Agency/Information Exploitation Office (DARPA/IXO)
3701 N. Fairfax Drive
Arlington, VA 22203-1714

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL-VA-WP

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TM-2006-3187

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Report contains color. PAO Case Number: AFRL/WS 06-1993, 16 Aug 2006.

14. ABSTRACT

- Generalize recent results in "Reachability of objectives" in discrete probablistic games to hybrid systems
- Develop algorithms and tools for estimating safe/desired performance for multi-modal and multi-vehicle systems
- Develop fault detection/handling tools and Markov-based decision process tools for incompletely observed modules.

15. SUBJECT TERMS
hybrid systems, multi-modal, software enabled control

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Raymond A. Bortner
Unclassified	Unclassified	Unclassified	SAR	62	**19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

TABLE OF CONTENTS

Page

Chapter One: Introduction--- 1

Chapter Two: Accomplishments in 2000------------------------------------- 4

Chapter Three: Accomplishments in 2001----------------------------------- 10

Chapter Four: Accomplishments in 2002------------------------------------- 17

Chapter Five: Accomplishments in 2003-------------------------------------- 25

Chapter Six: Accomplishments in 2004--------------------------------------- 35

Chapter Seven: Experimentation--- 40

Appendix: List Of Publications-- 43

Chapter One: Introduction

This Software Enabled Control project, sponsored by DARPA, was a landmark project at Berkeley in several different ways. It supported the research of three faculty investigators: Shankar Sastry, Tom Henzinger and Edward Lee and several graduate students (by our estimate over 30 graduate students over the life of the project) and several post doctoral scholars. Some examples of the project alumni and their current positions are

- George Pappas, University of Pennsylvania
- Rene Vidal, Johns Hopkins University
- Jin Kim, Seoul National University
- Omid Shakernia, Northrop Grumman
- Judy Liebmann, Lawrence Livermore National Laboratories
- Benjamin Horowitz, Lawrence Livermore National Laboratory
- Cedric Ma, Northrop Grumman
- John Koo, Vanderbilt University
- Maria Prandini, University of Milano, Italy
- Karl Johann Johansson, Royal Institute of Technology, Stockholm, Sweden
- Christoph Kirsch Meyer, University of Salzburg, Austria
- Steven Neundorffer, Xilinx Corporation
- Luca De Alfaro, University of California, Santa Cruz
- Rupak Majumdar, University of California- Los Angeles
- Jie Liu, Microsoft Research Corporation
- Lara Crawford, PARC Research Center, Palo Alto

Major Research in the project fell into three areas:

Hybrid and Multi-Modal Systems

The full theory of hybrid systems and safe design of controllers have been developed for safety and liveness specifications for hybrid systems. Work has been pioneered on both deterministic and stochastic hybrid control systems and developed analysis techniques. An annual workshop called "Hybrid Systems: Computation and Control" has been started, which is now the leading workshop on the topic in the world.

Design of Embedded Systems and High Confidence Systems

We developed time triggered and asynchronous techniques for the synthesis of hybrid systems. We developed tools for use with Ptolemy such as HyVisual and other tools such as Giotto and Massacio for designing time triggered systems. We began a workshop series called Embedded Software (Emsoft) which has become the leading workshop in the world for the topic

Experimental Work

We demonstrated on our helicopter platform during the course of the project numerous innovative projects using the methods developed during the course of our research:

Landing on pitching decks
Reinforcement Learning based control
Acrobatic Maneuvers using reinforcement learning
Conflict Detection and Resolution
Formation Flying
Air to Air Combat
Map Building and obstacle avoidance

A number of these approaches were transitioned to other DARPA programs such as UCAR, the Perch and Move program, the A-160 Maverick transition program. Additionally our experimental work enabled us to transition the results to the Boeing Open Control Platform for the Capstone Demo in August 2004.

Accomplishments from 1999-2004

We considered the design problem of embedded software for multi-vehicle multi-modal systems. Motivated by our design experience on the development of embedded software for our helicopter-based unmanned aerial vehicles which are composed of heterogeneous components, we believe that at the level closest to the environment under control, the embedded software needs to be time-triggered for guaranteed safety; at the higher levels, we advocate asynchronous hybrid controller design. Since the simulation is a precursor to the implementation of the control laws on real application. We will use a heterogeneous model simulator for high-level control designs. This will be done in a way consistent with and in support of the DARPA Open Control Platform (OCP), in order to bridge the gap between functional control design and software implementation. Work focused on the realization of the system for hardware-in-the-loop (HIL) simulation. HIL simulation allows testing an embedded computing device by progressively replacing the real-world environment at the device's input-output interface with a simulated environment. HIL simulation facilitates repeatable testing, as well as quantification of design tradeoffs. We began the construction of a HIL simulation system for a group of unmanned aerial vehicles (UAVs). We also began planning for mid-term demonstration is to fly hybrid control software on the Berkeley helicopters. Then, in the final demonstration, we planned to use the OCP publish-and-subscribe architecture for mutli-vehicle coordination. By way of abbreviations, MMCC stands for MultiModal Cooperative Control, HCSS is High Confidence Software and Systems.

Chapter Two: Accomplishments in 2000

We extended the theory and implementation of the model checker HyTech to cope with rectangular and nonlinear hybrid systems. We also developed asystematic classification of which requirements of hybrid systems can be model checked. Finally, we started a new project, FRESCO, on faithfully implementing hybrid models in real-time software.

Beyond HyTech: Hybrid Systems Analysis Using Interval Numerical Methods

Since hybrid embedded systems are pervasive and often safety-critical, guarantees about their correct performance are desirable. The hybrid systems model checker HyTech provides such guarantees and has successfully verified some systems. However, HyTech severely restricts the continuous dynamics of the system being analyzed and, therefore, often forces the use of prohibitively expensive discrete and polyhedral abstractions. Ben Horowitz, Rupak Majumdar, and Howard Wong-Toi designed a new algorithm, which is capable of directly verifying hybrid systems with general continuous dynamics, such as linear and nonlinear differential equations. The new algorithm conservatively over-approximates the reachable states of a hybrid automaton by using interval numerical methods. Interval numerical methods return sets of points that enclose the true result of numerical computation and, thus, avoid distortions due to the accumulation of round-off errors. The new algorithm is implemented in a successor tool to HyTech called HyperTech. Three examples are considered: a thermostat with delay, a two-tank water system, and an air-traffic collision avoidance protocol. HyperTech enables the direct, fully automatic analysis of these systems,which is also more accurate than the use of polyhedral abstractions.

Robust Undecidability of Timed and Hybrid Systems

The algorithmic approach to the analysis of timed and hybrid systems is fundamentally limited by undecidability, of universality in the timed case (where all continuous variables are clocks), and of emptiness in the rectangular case (which includes drifting clocks). Traditional proofs of undecidability encode a single Turing computation by a single timed trajectory. These proofs have nurtured the hope that the introduction of "fuzziness" into timed and hybrid models (in the sense that a system cannot distinguish between trajectories that are sufficiently similar) may lead to decidability. J-F Raskin showed that this is not the case, by sharpening both fundamental undecidability results. Besides the obvious blow our results deal to the algorithmic method, they also prove that the standard model of timed and hybrid systems, while not "robust" in its definition of trajectory acceptance (which is affected by tiny perturbations in the timing of events), is quite robust in its mathematical properties: the undecidability barriers are not affected by reasonable perturbations of the model.

Symbolic Model Checking for Rectangular Hybrid Systems

An important case of hybrid systems are the rectangular automata. First, rectangular dynamics can naturally and arbitrarily closely approximate more general, nonlinear dynamics. Second, rectangular automata are the most general type of hybrid systems for which model checking --in particular, LTL model checking-- is decidable. However, on one hand, the original proofs of decidability did not suggest practical algorithms and, on the other hand, practical symbolic model-checking procedures --such as those implemented in HyTech-- were not known to terminate on rectangular automata. Rupak Majumdar remedied this unsatisfactory situation: he developed a symbolic method for LTL model checking which can be performed by HyTech and is guaranteed to terminate on all rectangular automata. This is done by proving that our method for symbolic LTL model checking terminates on an infinite-state transition system if the trace-equivalence relation of the system has finite index, which is the case for all rectangular automata.

A Classification of Symbolic Transition Systems

Rupak Majumdar defined five increasingly comprehensive classes of infinite-state systems, called STS1-5, whose state spaces have finitary structure. For four of these classes, examples from hybrid systems can be provided.

STS1: These are the systems with finite bi-similarity quotients. They can be analyzed symbolically by (1) iterating the predecessor and Boolean operations starting from a finite set of observable state sets, and (2) terminating when no new state sets are generated. This enables model checking of the mu-calculus.

STS2: These are the systems with finite similarity quotients. They can be analyzed symbolically by iterating the predecessor and positive Boolean operations. This enables model checking of the existential and universal fragments of the mu-calculus.

STS3: These are the systems with finite trace-equivalence quotients. They can be analyzed symbolically by iterating the predecessor operation and a restricted form of positive Boolean operations (intersection is restricted to intersection with observables). This enables model checking of linear temporal logic.

STS4: These are the systems with finite distance-equivalence quotients (two states are equivalent if for every distance d, the same observables can be reached in d transitions). The systems in this class can be analyzed symbolically by iterating the predecessor operation and terminating when no new state sets are generated. This enables model checking of the existential conjunction-free and universal disjunction-free fragments of the mu-calculus.

STS5: These are the systems with finite bounded-reachability quotients (two states are equivalent if for every distance d, the same observables can be reached in d or fewer transitions). The systems in this class can be analyzed symbolically by iterating the predecessor operation and terminating when no new states are encountered. This enables model checking of reachability properties.

FRESCO: Formal Real-time Software Components

We have been designing a formal, structured modeling language for hybrid control systems, called Masaccio, and a time-triggered implementation language, called Giotto. Our eventual goal is to establish requirements in Masaccio by model checking, in a way which guarantees that the Giotto implementation meets the requirements.

Embedded Software Design

We have been experimenting extensively with embedded computing devices, sensors, and actuators, to gain experience and to identify a suite of experimental platforms that are safer to experiment with than the Bear Project helicopters. The objective is to develop techniques for portable, component-based design of embedded software. We plan to use the same techniques for programming a variety of platforms, including, eventually, the autonomous helicopters. We have performed quite a few investigations, and made quite a bit of progress in understanding this space.

Jini and Java Spaces

To give us a versatile experimental platform in anticipation of the Boeing OCP, we developed a publish-and-subscribe mechanism in Ptolemy II based on Java Spaces, a Linda-style distributed messaging system from Sun. Jie Liu and Yuhong Xiong created a demonstration where a publisher extracts stock market data from a web source and and subscriber consumes that data to generate trend plots. This demonstration shows the network integration and publish-and-subscribe interactions working with Ptolemy II. Moreover, the demo used Jini, a networked service-discovery mechanism from Sun, intended for embedded devices. Service discovery is not currently part of the OCP plan, but we hope to develop a better understanding of its role to help determine whether it should be incorporated later, and in what form.

Pioneer and Jini

Jie Liu and Steve Neuendorffer created a software infrastructure in Java for programming a Pioneer mobile robot borrowed from the Bear project and the Legos using the same code. This is our first serious demonstration of platform independent software development. The Pioneer is a much more substantial platform than the Legos, and has a single-board computer running Linux. Their software uses Jini to distribute code to robots, or in the case of the Legos, to a proxy for the robots running on a laptop computer. The code that is distributed is Java bytecode. When a task is needed of a robot, an instance of a Java class is posted to the Jini server, and the robots are notified. Any robot that is available and implements an interface compatible with the posted task can grab the code from the Jini server to execute the task.

Robot Arm

Chamberlain Fong assembled from a kit a small robot arm with four degree of freedom motion and a gripper. He built a demonstration of a controller which this arm that uses a new domain in Ptolemy II that he is developing, DT (discrete time). The demonstration includes an on screen animation, through which commands are entered using the keyboard, and the mouse is used for changing the viewpoint of the arm simulator. The communication between software components is mostly angles and polygons generated by a forward kinematics controller. Chamberlain is working on an inverse kinematics controller. The robot arm is connected to a host computer through the serial port.

Handspring Visors

Christopher Hylands experimented with the KVM (a minimal Java virtual machine) running on Handspring Visors (handheld electronic organizers). We selected the Handspring Visor instead of the more popular Palm Pilot because the Visor includes a bus with a published interface, and thus has much more promise for use in embedded systems. Christopher was able to get a small portion of the Ptolemy II code to compile for the KVM, and was able to port a small process networks application. Christopher attempted to get the IR port on the Handspring to communicate with the Legos, but was unsuccessful.

Dallas Tini

We obtained a pair of Dallas Seminconductor Tini boards, small single-board computers that include a Java virtual machine in a DIMM form factor. They support a serial port, a 1-wire interface, CAN bus and 10baseT ethernet.

http://www.ibutton.com/TINI/software/index.html says:

"The Java VM on TINI conforms to Sun's *Embedded* Java platform, version 1.1 of the Java API. Embedded in TINI's flash memory are java.lang, java.net, java.io, java.util; there is room in RAM for other packages of your choice. Also in flash memory are com.dalsemi packages for accessing the TINI command shell slush, the 1-Wire® bus and several system parameters."

TINI implements none of the serialization, reflection, and dynamic class loader, so it is still far from RMI. That is, at this time, it is impossible to make the TINI board a Jini device and access the Java Space directly. It is doubtful that these devices would be easy to interface to the OCP, except through proxies running on a more substantial host. Nonetheless, there are interesting possibilities.

Dallas makes a small memory device called an iButton. The idea is that you might carry an iButton with you (attached to a

ring, or a key, or a badge) and the Tini board reads it and personalizes its environment to you.

Real-Time Java

Because the extensive Java infrastructure that has been built (in the form of Ptolemy II), we have been studying the various real-time Java projects with an eye towards being able to provide native Java support for the OCP, including its quality of service features. Particular realizations we have studied include Esmertec, FIXME.

CORBA Sensors

Brian Vogel created a demonstration that uses CORBA to obtain and plot data from an ultrasonic detector connected to a Windows NT machine running a web server. He uses the audio hardware of the PC as an analog interface to the sensor. This was our first networked sensor application, and it illustrates how smart sensors might communicate with controllers via the OCP.

Controller Design and Analysis

We did extensive research on hybrid system theory which enable multi-agent multi-modal control. The objective is to use hybrid system model for the design, analysis and synthesis of multi-agent multi-modal systems that deliver high levels of mission reliability in dynamic and rapidly evolving environments.

Multi-Modal Control Derivation

John Koo, Shankar Sastry, and George Pappas of University of Pennsylvania worked on mode switching synthesis for specifications which are based on mission completion properties. In particular, we were interested in reachability specifications. The objective was to develop a general framework for the study of control mode switching and algorithms for the derivation of sequences of control modes which will satisfy reachability tasks. We implemented the algorithms based on existing efficient methods for reachability computation and apply the algorithms to certain classes of dynamical systems.

Component-based Design Technique for Multi-Modal Control

Based on previous works by Jie, Xiaojun, John Koo and Bruno Sinopoli on the implementation of a multi-modal system with fixed mode sequences as a hierarchical hybrid system in Ptolemy II, we investigated the use of the proposed component-based design technique to implement a scaled down version of UAV flight management system which facilitates on-line control mode sequence generation. The proposed mode switching algorithm for reachability specifications will be used to demonstrate the versatility of the proposed software design technique for the design of embedded software which enables multi-modal control.

Computations for Optimal Hybrid Control

Rene Vidal, Shawn Schaffert, John Lygeros and Shankar Sastry have developed algorithms for computing the maximal controlled invariant set and least restrictive controller for discrete time systems. We show that the algorithm can be encoded using quantifier elimination, which leads to a semi-deciablity result for definable system. Rene Vidal and Omid Shakernia have implemented an algorithm for a least restrictive controller synthesis based on robust semidefinite programming. Omid Shakernia has shown theoretical result on decidable controller synthesis for classes of linear systems.

Modeling of Hybrid Systems

The objective of the work was to have a deeper and border understanding of dynamical behaviors of hybrid systems in order to benefit the development of new algorithms for analysis, synthesis, verification and simulation of hybrid systems. Jun Zhang, Karl Johansson, John Lygeros and Shankar Sastry have extended several important results of classical dynamical systems to hybrid dynamical systems, especially those with Zeno executions. Slobodan Simic, Karl Johansson, John Lygeros and Shankar Sastry have introduced a new framework for a global geometric study of hybrid systems. Its usefulness is demonstrated in analysis of the Zeno phenomenon and stability of hybrid equilibria. Ekaterina Lemch has studied the controllability of hybrid system in this geometric framework.

Probabilistic Hybrid Systems

Fault detection and handling need to be designed so as to guarantee a hierarchical decrease in functionality from the least safety-critical to the most. Off-line, the fault handling routines need to be analyzed and verified in a probabilistic framework. Jianghai Hu, John Lygeros and Shankar Sastry have extended the deterministic framework of hybrid system to the nondeterministic one. This provides an analytical framework for the
verification of fault detection and handling.

UAV Control Software Development

David Shim, Xiaojun and John Koo has been working on formal modeling of existing embedded software into Unified Modeling Language. The objective is to capture the organization of software components, scheduling scheme provided by operating systems and temporal properties and constraints of different hardware and software components.

Chapter Three: Accomplishments in 2001

Our effort integrated three tasks: formal modeling and verification; embedded software design; and controller design and analysis. On the formal side, we introduced a hierarchical model for the design and analysis of embedded control systems called Masaccio. This will be the basis for the next generation of the hybrid model checker Hytech. We also focused heavily on the study of games for synthesizing controllers. Our original results concern the solution of probabilistic games (for modeling synchronous control and uncertain environments) and abstraction mechanisms for games. At the PI meeting in Albuquerque, we demonstrated hierarchical heterogeneous modeling and design using Ptolemy II, with integrated smart sensors and motor-drive actuators to control small robots. We also described a preliminary implementation of a time-triggered language, and progress on its underlying formalism. On the control side, we showed that the controller synthesis problem is semi-decidable for special classes of hybrid systems where the continuous vector fields are linear, and we obtained original results in the optimal control of switched linear systems.

Formal Modeling and Analysis

We defined a hierarchical model for the design and analysis of embedded control systems called Masaccio. The next generation of Hytech is planned to operate on Masaccio models. We also focused heavily on the study of games. Games are the natural setting for studying the synthesis of controller: a controller is a winning strategy of a game system vs. control. Our original results concern the solution of probabilistic games (for modeling synchronous control and uncertain environments) and abstraction mechanisms for games. To our knowledge, our work on abstraction of games is the first attempt at an abstract interpretation theory for controller synthesis.

Masaccio: a formal model for embedded components

We defined Masaccio, a formal model for hybrid dynamical systems which are built from atomic discrete components (difference equations) and atomic continuous components (differential equations) by parallel and serial composition, arbitrarily nested. Each system component consists of an interface, which determines the possible ways of using the component, and a set of executions, which define the possible behaviors of the component in real time. The purpose of Masaccio is to enable hierarchical formal modeling and verification of hybrid systems.

Concurrent omega-regular games

We consider two-player games which are played on a finite state space for an infinite number of rounds. The games are concurrent, that is, in each round, the two players choose their moves independently and simultaneously; the current state and the two moves determine a successor state. We consider omega-regular winning conditions on the resulting infinite state sequence. To model the independent choice of moves, both players are allowed to use randomization for selecting their moves. This gives rise to the following qualitative modes of winning, which can be studied without numerical considerations concerning probabilities: sure-win (player 1 can ensure winning with certainty), almost-sure-win (player 1 can ensure winning with probability 1), limit-win (player 1 can ensure winning with probability arbitrarily close to 1), bounded-win (player 1 can ensure winning with probability bounded away from 0), positive-win (player 1 can ensure winning with positive probability), and exist-win (player 1 can ensure that at least one possible outcome of the game satisfies the winning condition).

We provide algorithms for computing the sets of winning states for each of these winning modes. In particular, we solve concurrent Rabin-chain games in n^O(m) time, where n is the size of the game structure and m is the number of pairs in the Rabin-chain condition. While this complexity is in line with traditional turn-based games, where in each state only one of the two players has a choice of moves, our algorithms are considerably more involved than those for turn-based games. This is because concurrent games violate two of the most fundamental properties of turn-based games. First, concurrent games are not determined, but rather exhibit a more general duality property which involves multiple modes of winning. Second, winning strategies for concurrent games may require infinite memory.

Abstract interpretation of game properties

We apply the theory of abstract interpretation to the verification of game properties for reactive systems. Unlike properties expressed in standard temporal logics, game properties can distinguish adversarial from collaborative relationships between the processes of a concurrent program, or the components of a parallel system. We consider two-player concurrent games ---say, component vs. environment--- and specify properties of such games ---say, the component has a winning strategy to obtain a resource, no matter how the environment behaves--- in the alternating-time mu-calculus (AMU). A sound abstraction of such a game must at the same time restrict the behaviors of the component and increase the behaviors of the environment: if a less powerful component can win against a more powerful environment, then surely the original component can win against the original environment.

We formalize the concrete semantics of a concurrent game in terms of controllable and uncontrollable predecessor predicates, which suffice for model checking all AMU properties by applying boolean operations and iteration. We then define the abstract semantics of a concurrent game in terms of abstractions for the controllable and uncontrollable predecessor predicates. This allows us to give general characterizations for the soundness and completeness of abstract games with respect to AMU properties. We also present a simple programming language for multi-process programs, and show how approximations of the maximal abstraction (w.r.t. AMU properties) can be obtained from the program text. We apply the theory to two practical verification examples, a communication protocol developed at the Berkeley Wireless Research Center, and a protocol converter. In the wireless protocol, both the use of a game property for specification and the use of abstraction for automatic verification were instrumental to uncover a subtle bug.

Embedded Software Design

The objective of this part of the project is to adapt sophisticated software techniques to the problem domain of real-time control systems. The problem domain mandates a priority on real-time guarantees and verifiable designs. Our approach is based on formalizing the models of computation that govern the interaction between components (both hardware and software components), and building software that realizes these models of computation. Control systems integrate a variety of hardware, including networked sensors and actuators, with variety of service demands, ranging from hard-real-time signal processing to interaction with human operators. Hence, our approach emphasizes the composition of heterogeneous models of computation.

For example, publish-and-subscribe interactions between components (with priorities) are suitable for managing irregular events and alarms, but not so well suited for regular, high-sample-rate signal processing nor for continuous dynamics. Time-triggered formalisms are better suited for hard-real-time processing, but not as well suited for irregular interactions. Our approach, therefore, is to study the composition of these models and to build software that enables systematic composition. Another example of such heterogeneous compositions uses state machine models to govern modes of operation of systems whose behavior is better represented in some other way, such as by continuous-time models, to get hybrid systems.

We are applying modern software concepts, including sophisticate approaches to concurrency management, discovery, middleware, type systems, and higher-order functions to this problem domain. We are using a pre-existing Java-based software framework called Ptolemy II as an experimental platform, and we are working with the Berkeley Aerobots project to put our software concepts on autonomous model helicopters.

Fresco

Ben Horowitz and Christoph Meyer have finished a draft implementation of a skeleton of Giotto in VxWorks. Giotto is our evolving time-triggered implementation language. We intend the implementation to be portable between RTOSs (e.g., Real-time Linux, QNX, etc.) In particular, we have limited our use to POSIX features. We found the logic analyzer of VxWorks very helpful, both for debugging and for understanding the behavior of the RTOS.

There is a screen shot available at

http://www-cad.eecs.berkeley.edu/~fresco/teos/teos3-log-2.gif,

which serves as a good example. In it, one can see four processes of interest:

- com, which is responsible for intertask communication
- task0, which executes four times per round
- task1, which executes two times per round
- task2, which executes once per round

There is also an idle task, which is not of interest. Here's what's going on:

- at time 12, the communication task begins to run. It makes task0, task1, and task2 ready to run. The heavy green horizontal line signifies that a task is running, and the sawtooth horizontal line signifies that a task is ready.
- at time 22, task0 begins to run, and finishes shortly thereafter.
- at time 23, task1 begins to run. It finishes at time 34.
- at time 34, task2 begins to run. It gets preempted by com at time 43.
- at time 50, task1 begins to run, and finishes shortly thereafter.
- at time 52, task2 resumes, and finishes at time 61.
- ...

Note that task2 continues to run across multiple invocations of task0. This is possible because of the preemptive scheduler of VxWorks.

Albuquerque Demo

At the PI meeting in Albuquerque, we demonstrated hierarchical heterogeneous modeling and design using Ptolemy II to integrate smart sensors and motor-drive actuators to control small robots. The implementation used a publish-and-subscribe fabric (JavaSpaces), a service discovery fabric (Jini), a discrete-event simulation engine (the Ptolemy II DE domain), and a dataflow computation engine (the Ptolemy II SDF domain). It showed models that were modified during execution, with thread-safe updates of static analysis information. The discovery mechanism was used to dynamically discover the publish and subscribe fabric. We believe that the same infrastructure can be used to discover other publish-and-subscribe fabrics, such as the OCP.

The small robots were Lego Mindstorms, and the sensor was a tilt sensor from Telemonitor connected via an IEEE 1451.2 bus to an Agilent NCAP (network capable application processor), which hosted a web server. The robots and the sensors were abstracted as Ptolemy II actors. In the case of the robots, the actor functioned as a proxy, internally implemented by sending commands to the robot over the serial port and an infrared link. To improve reliability and directionality of the infrared link, we used a port replicator (designed and built by Win Williams) to drive two distinct infrared emitters. The Telemonitor sensors were abstracted as actors that performed HTTP queries of the NCAP to obtain sensor information and to configure the sensors.

A number of issues were raised by this demo, particularly with regard to the semantics of the publish-and-subscribe infrastructure. There are a number of options in configuring this infrastructure. For instance:

- Should events have time stamps?
- Should these time stamps have global significance?
- Should delivery of events be reliable (with retry if necessary)?
- Should events always be delivered in the order they are produced?
- How can we coordinate events from multiple sources?
- Should synchronous event delivery be supported?
- How should events from multiple publishers be merged?
- How should dynamic redirection and resourcing of events be supported?
- Should the fabric provide persistent state?
- Should the fabric provide history services?

In view of these questions, the next step is to define a precise semantics for a publish-and-subscribe fabric for software-enabled control systems.

The demo builders were Chamberlain Fong, Christopher Hylands, Jie Liu, Xiaojun Liu, Steve Neuendorffer, Sonia Sachs, and Win Williams.

Discovery

At the Albuquerque PI meeting, we also proposed the use of discovery mechanisms (such as Jini) to construct a "Meta-OCP," where various configurations of interconnection services could be delivered in a modular way. For example, a component may request a reliable stream-based delivery mechanism to get sampled data from point A to point B. The meta-OCP would respond with byte-code that implements a suitable interface, using for example TCP/IP and sockets, bypassing any central infrastructure. This might be used to transport regularly sampled data, such as audio data. In another example, a component requests a shared data repository visible to a number of components. The delivered code might interact with a Linda-style tuple space, such as that in the OCP. This might be used, for example, to read the current temperature from a sensor. In a third example, a component requests an interface to send time-stamped data that must be delivered and dealt with within some specified deadline. The delivered code might interact with TAO via the OCP. This might be used, for example, to deliver motion control data.

Controller Design and Analysis

We have done extensive research on hybrid system theory which enable multi-agent multi-modal control. The objective is to use hybrid system model for the design, analysis and synthesis of multi-agent multi-modal systems that deliver high levels of mission reliability in dynamic and rapidly evolving environments. For controller synthesis, we have worked on multi-modal control derivation and computations for optimal hybrid control. The objective of multi-modal control research is to develop a general framework for the study of control mode switching and algorithms for the derivation of sequences of control modes which will satisfy reachability tasks. Synthesis of least restrictive controllers based on optimal control theory for classes of systems in which the reachability problem are deciable are undertaking. Modeling of hybrid systems in both deterministic and nondeterministic settings are currently being investigated. The objective of the studies is to have a deeper and border understanding of dynamical behaviors of hybrid systems in order to benefit the development of new algorithms for analysis, synthesis, verification and simulation. Component-based design technique will be deployed for the design of embedded software which enables multi-modal control. In parallel, formal modeling language, unified modeling language, is used for modeling of existing embedded software running on helicopter-based unmanned aerial vehicle developed at Berkeley to motivate the study in applying component-based design techniques for real-time embedded system.

Semi-decidable Controller Synthesis for Classes of Linear Hybrid Systems

A problem of great interest in the control of hybrid systems is the design of least restrictive controllers for reachability specifications. Controller design typically uses game theoretic methods to compute the region of the state space for which there exists a control such that for all disturbances, an unsafe set is not reached. In general, the computation of the controllers requires the steady state solution of a Hamilton-Jacobi partial differential equation which is very difficult to compute, if it exists. We show that for special classes of hybrid systems where the continous vector fields are linear, the controller synthesis problem is semi-decidable: There exists a computational algorithm which, if it terminates in a finite number of steps, will exactly compute the least restrictive controller. This result is achieved by a very interesting interaction of results from mathematical logic and optimal control and presents us with the first semi-decidable controller synthesis result for such rich classes of hybrid systems.

Toward Optimal Control of Switched Linear Systems

The problem of driving the states of a switched linear control system between boundary states is investigated. We propose tight lower bounds for the minimum energy control problem. We furthermore show that the fact that the dynamics is changing discontinuously on the switching surface gives rise to phenomena that can be treated as a decidability problem for hybrid control systems, using ideas from mathematical logic.

Chapter Four: Accomplishments in 2002

Giotto

We continued development on the time-triggered Giotto model of computation. Specifically, we are investigating the schedulability of Giotto on distributed platforms (Ben Horowitz) and we have implemented a Giotto compiler on a UAV rotorcraft from ETH Zurich (Christoph Kirsch, and Marco Sanvido drom ETH). All ground tests of Giotto on the model helicopter have been successfully completed and we are currently awaiting some sort of certification for flying. Giotto implements the glue (scheduling) code for the entire flight control systems, which manages the execution of native software tasks written in Oberon.

The first Giotto paper was published in the ACM Workshop for Languages, Compilers, and Tools for Embedded Systems: Giotto is a principled, tool-supported design methodology for implementing embedded control systems on platforms of possibly distributed sensors, actuators, CPUs, and networks. Giotto is based on the principle that time-triggered task invocations plus time-triggered mode switches can form the abstract essence of programming real-time control systems. Giotto consists of a programming language with a formal semantics, and a retargetable compiler and runtime library. Giotto supports the automation of control system design by strictly separating platform-independent functionality and timing concerns from platform-dependent scheduling and communication issues. The time-triggered predictability of Giotto makes it particularly suitable for safety-critical applications with hard real-time constraints.

Dynamic Programming Algorithms for Controller Synthesis

Symbolic model checking algorithms for system verification can be viewed as dynamic programs. We showed that the same dynamic programming schemes can also be used for the automatic synthesis of optimal control strategies. This work lies the foundation for using a model-checking tool such as HyTech for controller synthesis. We published these results at the 2001 IEEE Symposium on Logic in Computer Science.

Dynamic programs, or fixpoint iteration schemes, are useful for solving many problems on state spaces, including model checking on Kripke structures ("verification"), computing shortest paths on weighted graphs ("optimization"), computing the value of games played on game graphs ("control"). For Kripke structures, a rich fixpoint theory is available in the form of the mu-calculus. Yet few connections have been made between different interpretations of fixpoint algorithms. We study the question of when a particular fixpoint iteration scheme f for verifying an omega-regular property L on a Kripke structure can be used also for solving a two-player game on a game graph with winning objective L. We provide a sufficient and necessary criterion for the answer to be affirmative in the form of an extremal-model theorem for games: under a game interpretation, the dynamic program f solves the game with objective L if and only if both (1) under an existential interpretation on Kripke structures, f is equivalent to EL, and (2) under a universal interpretation on Kripke structures, f is equivalent to AL. In other words, f is correct on all two-player game graphs iff it is correct on all extremal game graphs, where one or the other player has no choice of moves. The theorem generalizes to quantitative interpretations, where it connects two-player games with costs to weighted graphs.

While the standard translations from omega-regular properties to the mu-calculus violate (1) or (2), we give a translation that satisfies both conditions. Our construction, therefore, yields fixpoint iteration schemes that can be uniformly applied on Kripke structures, weighted graphs, game graphs, and game graphs with costs, in order to meet or optimize a given omega-regular objective.

E-Machine

We have initiated an effort to define a "virtual machine" layer that abstracts the real-time and task interaction properties of an embedded system. The intent of the virtual machine is to serve as a target for a problem-oriented "controls API," and to be implementable on a variety of hardware/software architectures, including the current Boeing OCP.

The virtual machine layer concept is our proposal to reconcile the needs of distributed supervisory control, which matches a publish and subscribe style of interaction using real-time CORBA, with lower-level hard-real-time control, which matches time-driven models such as FRP (from Yale) and Giotto (from Berkeley). At the same time, the virtual machine is attempting to address issues of concurrency management, particularly with respect to precise reactions and precise mode switching, in a way that is independent of the level of control being applied. Thus, the same conceptual framework can be shared.

A first attempt at this virtual machine definition has been carried out by Christoph Kirsch and Tom Henzinger, and is described at:

http://www.eecs.berkeley.edu/~fresco/giotto/refs

Their virtual machine is called the E-machine, for "embedded machine." A program in this virtual machine consists of sets of tasks that take time and cannot be logically interrupted. There are no synchronization points between these tasks, thus avoiding a key source of problems with deadlock and unexpected nondeterminacy in concurrent programs. Instead, interactions between tasks are through ports. Input ports provide values that do not change during execution of the task. Output ports are updated only at the completion of the task. This is key to the operation of the virtual machine, and provides the principle mechanism for achieving predictable behavior.

Tasks take time to execute, and execute concurrently with other tasks. The virtual machine does not define the underlying task scheduling model. That scheduling model would typically be provided by an underlying real-time OS. A dispatcher enables tasks for execution, and runs at the highest priority. The only assumption made about the underlying task scheduling model is that if the dispatcher has work to do, then it will execute, preempting all other tasks.

In addition to tasks that take time, the virtual machine defines "drivers," which are tasks that conceptually execute instantaneously. These execute at the priority of the dispatcher.

Time information is provided by the environment through ports, and these ports can trigger the enabling of tasks. Ports can also be used to signal the completion of a task, and activation of tasks can depend on port values.

An E machine is an interpreter for E code. A compiler could generate the E machine interpreter, including the task manager (RTOS generation).

Jie Liu has developed a proposal for a distributed E machine interpreter built using the Boeing OCP. This is an evolving proposal, and discussions with Boeing are under way.

Virtual Machine Working Group
On June 13, we held an all day meeting at Berkeley to get inputs from the technology developers in SEC on the virtual machine concept. Attending were: John Peterson, Yale, Richard Kieburtz, OGI, Jie Liu, Berkeley, Walid Taha, Yale, Johan Nordlander, OGI, Tom Henzinger, Berkeley, Edward A. Lee, Berkeley, Ben Horowitz, Berkeley, Magnus Carlsson, OGI, Joern Janneck, Berkeley, and Xiaojun Liu, Berkeley

Jie Liu gave an overview of the OCP, and we discussed a virtual machine interpretation of the OCP. Johan Nordlander described the object model underlying O'Haskell (now called Timber), which we identified as a programming model rather than a virtual machine. Magnus Carlsson described an alternative to the OCP event channel for task interaction. Tom Henzinger described his strawman virtual-machine design, and argued that it offers to the virtual machine concept several key ideas:

- atomic execution ("drivers")
- ability to have a distributed time model.
- Determinism in two dimensions:

  - Time it takes to do something (delay committing results)
  - Values of inputs and outputs (double buffering)

Jie Liu described our hierarchical preemptive multitasking (HPM) domain in Ptolemy II, and we identified a realization of this domain on the strawman virtual machine. This domain offers a solution to the precise mode change problem in the OCP, and shares with the E-machine the split phase read/commit of inputs and outputs.

John Peterson described FRP, which he identified as low-level semantics work. FRP adds time flow to any existing language. It is possible that FRP could be used to represent the semantics of the virtual machine. Walid Taha described real-time FRP, an abstract subset on FRP with guaranteed bounds on execution time and space.

We identified the following action items:

- Sell the split phase execution concept.
  - Solves the precise mode change problem.
  - Introduces a measure of determinism.
- Define the embedded machine
  - With a better syntax
  - Top-level control loop
  - An informal operational semantics
  - Define the task model
  - Type system (not in the critical path)
  - Distribution model (& connection to event channel model)
  - Prototype implementation
- Show how InputDataAvailable() can be converted to a split phase execution.
  - Backward compatible version might assume this is a driver.
- Show have various programming models map onto the EM.
  - Concurrent real-time FRP
  - HPM (hierarchical preemptive multitasking)
  - Giotto
  - Priority-driven publish/subscribe
  - Timber
  - Rate monotonic scheduling

Blending Controllers

Linda Wills and Bonnie Heck of Georgia Tech have been working on "blending controllers," where during a mode change, the control laws of two controllers are blended in a controlled way during the transition between the two controllers. These are modal (hybrid) controllers where instead of abruptly switching from one control law to another, an intermediate "blending" mode is use to gradually turn over control from one to the other.

We began working with them on exploring ways of realizing these controllers, first via conference calls, and then when Linda Wills and Murat Guller visited us on 6/5/01 and 6/6/01. They posed the question of how to identify a suitable programmer's model for such modal controllers. We explored various alternatives, including mechanisms that could be built immediately in Ptolemy II, but found none of them suitable. We then discovered that a small change to the FSM (finite state machine) domain in Ptolemy II would enable a very clean programmer's model. In this model, a three state FSM is defined where in the first state, controller A is active and in the third state, controller B is active. In the second state, both controllers are active and their outputs are blended using an expression. The small change required in the FSM domain is that states must be able to have more than one active refinement. This turned out to be relatively easily to implement. Xiaojun Liu and Jie Liu completed an implementation in about one hour, and we constructed a simple demonstration system in about another half hour.

Generalized Modal Models

An outcome of the meeting with Georgia Tech (see above) was the realization that the Ptolemy II notion of a modal controller can be usefully generalized to one where a mode can have more than one (concurrent) refinement. The finite-state machine governing the mode changes can then be viewed as providing a mask, making some subset of the actors of a model visible to the director. This appears to be a significant increase in expressiveness over our prior *-charts formalism, which is a generalization of Statecharts and hybrid systems. Some possible applications of this formalism:

1) It could be used for managing redundant systems and doing fault detection and isolation.

2) It could be that heterochronous dataflow is a special case.

3) It could be viewed as a structured mutations approach.

Xiaojun Liu is exploring this concept further.

HPM (formerly RTOS) Domain

Jie Liu has made major progress on his model of computation supporting precise reactions in the context of preemptive multitasking. In the previous reporting period, we reported a Ptolemy II domain called the RTOS domain that modeled RTOS-style preemptive multitasking. Jie has elaborated the semantics of this domain in very interesting ways, and has come up with a new model of computation that is significantly different from the previous version. We are calling this domain the HPM domain, for hierarchical preemptive multitasking.

The key idea in HPM is that of split-phase firing, inspired by the Giotto model. In this manifestation, actors read and locally cache input data when their prefire() method is invoked, and (optionally) begin computation in a background process that executes concurrently with other processes with some specified priority. Not until the fire() method is invoked, however, do these actors write output data. In Jie's model, the actors declare their execution time, and the fire() method is invoked after they have had available CPU execution time equal to this declared value. A dispatcher process keeps track of invocations of actors with higher priority so that the process is assured of getting the declared CPU time. Since the outputs are not committed until that time has elapsed, there are several possible design patterns that can be overlayed on this:

* Terminator pattern. If a process is not ready to produce outputs when its time has elapsed, then it has violated its contract on declared execution time, and it is terminated, not to be invoked again.

* Anytime pattern. If a process has not completed its computation when its time has elapsed, then partial results are extracted and produced.

* Laisez faire pattern. When the fire() method is invoked, it performs a join with the executing process, allowing it to run to completion.

There are circumstances where each of these patterns is appropriate, and the choice depends on the application. It does not make sense to fix one of these choices in the modeling framework.

Continuous-Time Domain

With the help of Gautam Biswas and others at Vanderbilt University, Jie Liu identified limitations in the semantics of the CT domain (continuous time) when combined with finite state machines (FSMs) to make hybrid systems. Previously, actors were identified as either discrete or continuous, and the CT kernel treated the two differently. Any actor downstream from a discrete actor was also discrete. However, when building hybrid systems, it is common for an actor to interact with both discrete and continuous signals. Jie solved the problem by identifying signals instead of actors as discrete or continuous, and using a mechanism similar to type resolution to determine for each signal whether it is discrete or continuous.

High Confidence Control of Multi-Modal Systems

In multi-modal control paradigm, a set of controllers of satisfactory performance have already been designed and must be used. Each controller may be designed for a different set of outputs in order to meet the given performance objectives and system constraints. When such a collection of control modes is available, an important problem is to be able to accomplish a variety of high level tasks by appropriately switching between the low-level control modes. T. John Koo, George Pappas (University of Pennsylvania), and Shankar Sastry presented a framework for determining the sequence of control modes that will satisfy reachability tasks. Our framework exploits the structure of output tracking controllers in order to extract a finite graph where the mode switching problem can be efficiently solved, and then implement it using the continuous controllers. Our approach is illustrated on a robot manipulator example, where we determine the mode
switching logic that achieves the given reachability task.

Hierarchical System Architecture for Multi-Agent Multi-Modal Systems

T. John Koo presented a hierarchical system architecture for multi-agent multi-modal systems. The design principle for the construction of the hierarchy is based on bisimulation and therefore a higher-level system and a lower-level system are bisimilar. The layered system is designed to promote proof obligations so that system specification at one level of granularity conforms with system specification at another level and vice versa. The approach is illustrated on designing a system architecture for executing a mission of controlling a group of autonomous agents in the pursuit of multiple evaders.

Hierarchical Approach for Design of Multi-Vehicle Multi-Modal Embedded Software

Embedded systems composed of hardware and software components are designed to interact with a physical environment in real-time in order to fulfill control objectives and system specifications. T. John Koo, Judy Liebman, Cedric Ma, and S. Shankar Sastry addressed the complex design challenges in embedded software by focusing on predictive and systematic hierarchical design methodologies which promoted system verification and validation. First, they advocate a mix of top-down, hierarchical design and bottom-up, component-based design for complex control systems. Second, it is their point of view that at the level closest to the environment under control, the embedded software needs to be time-triggered for guaranteed safety; at the higher levels, we advocate an asynchronous hybrid controller design. We briefly illustrate our approach through an embedded software design for the
control of a group of autonomous vehicles.

Hardware-in-the-loop Simulation of Multi-Vehicle Multi-Modal Embedded Systems

To design embedded systems for the control of autonomous vehicles for collectively delivering high levels of mission reliability has been putting tremendous pressure on control and software designers in industry. Hardware-in-the-loop (HIL) simulation allows testing an embedded computing device by replacing the real-world environment at the device's input-output interface with a simulated environment. HIL simulation is particularly effective when normal testing is dangerous or costly. HIL simulation facilitates repeatable testing, as well as quantification of design tradeoffs. T. John Koo, Benjamin Horowitz, Judy Liebman, Cedric Ma, Ron Tal, Tom Henzinger, and Shankar Sastry started to look into the issues on the design of a HIL simulation system for a group of unmanned aerial vehicles (UAVs).

Decidable and Semi-decidable Controller Synthesis for Classes of DiscreteTime Hybrid Systems

Rene Vidal, Shawn Schaffert, Omid Shakernia, John Lygeros, Shankar Sastry presented classes of discrete time hybrid systems for which the classical algorithm for computing the maximal controlled invariant set and the least restrictive controller is computable and guaranteed to terminate in a finite number of iterations. We show how the algorithm can be encoded using quantifier elimination, which leads to a semi-decidability result for definable hybrid systems. For discrete time linear systems with linear constraints that are either controllable or nilpotent and have bounded disturbances, we show that the controlled invariance algorithm terminates in a number of iterations which is at most the dimension of the system. Both in the hybrid and in the linear case, our results are much more general than the corresponding ones for continuous time systems. Finally they show that for linear systems with ellipsoidal constraints, an approximated solution can be obtained using robust convex programming. They provide an example showing that our algorithm gives better estimations than other ellipsoidal methods
and is more efficient than the exact method for linear constraints.

Chapter Five: Accomplishments in 2003

Our effort integrated three tasks: formal modeling and verification; embedded software design; and controller design and analysis. We made substantial progress in all three directions. In addition in the past year we had a large number of successful hardware demonstrations on our Berkeley Aerobotics (BEAR) rotorcraft UAVs. Our group is arguably the most advanced UAV research group in the country and the following experiments performed under SEC were highlighted at the recent PI meeting:

1. Giotto Implementation of Platform Based UAV Controller
2. Formation Flying with 2 real and 7 virtual UAVs.
3. UAV chicken with 2 UAVs flying at each other
4. Conflict Resolution for multiple UAVs (software demo)
5. Dogfights with multiple close in UAVs in complex urban environments (software demo).

This was in addition to complex acrobatic maneuvers (demonstrated in the May 02 PI meeting), vision based landing (presented by Northrop Grumann as possible transition target to their FireScout at May 03 PI meeting), pursuit evasion games. In addition we showed some preliminary work on electric rotorcraft and low weight fixed wing UAVs (the BAT UAV).

The solution of control problems for probabilistic systems with uncertain environments.

Behavioral properties of control systems can be formalized as objectives in two-player games. Turn-based games model asynchronous interaction between the players (the system and its environment) by interleaving their moves. Concurrent games model synchronous interaction: the players always move simultaneously. Probabilistic moves can be used to model uncertainty in the environment and failure scenarios for the system. We found efficient reductions of concurrent probabilistic Buchi and co-Buchi games to turn-based games with Buchi condition and parity winning condition with three priorities, respectively.

Giotto-based and platform-based implementations of helicopter flight control systems

Giotto is a platform-independent language for specifying software for high-performance control applications. We developed a new approach to the compilation of Giotto. Following this approach, the Giotto compiler generates code for a virtual machine, called the E machine, which can be ported to different platforms. The Giotto compiler also checks if the generated E code is time safe for a given platform, that is, if the platform offers sufficient performance to ensure that the E machine semantics. We demonstrated the feasibility and benefits of Giotto-based control software development by re-implementing the autopilot system of an autonomously flying model helicopter. Giotto offers a clean separation between the platform independent concerns of software functionality and I/O timing, and the platform dependent concerns of software scheduling and execution. Functionality code such as code computing control laws can be generated automatically from Simulink models or, as in the case of this project, inherited from a legacy system. I/O timing code is generated automatically from Giotto models that specify real-time requirements such as task frequencies and actuator update rates.

Formation reconfiguration planning for autonomous vehicles
Given a group of autonomous vehicles, an initial configuration, a final configuration, a set of inter- and intra- vehicle constraints, and a time for reconfiguration, the Formation Reconfiguration Planning problem is focused on determining a nominal input trajectory for each vehicle such that the group can start from the initial configuration and reach its final configuration at the specified time while satisfying the set of inter- and intra- vehicle constraints. S. Zelinski, T. J. Koo, and S. Sastry were interested in solving the Formation Reconfiguration Planning problem for a specific class of systems and a particular form of input signals so that the problem can be reformulated as an optimization problem which can be solved more efficiently, especially for a large group of vehicles.

Design, Analysis, and Implementation of the Embedded Virtual Machine
The Embedded Machine is a virtual machine that mediates in real time the interaction between software processes and physical processes. It separates the compilation of embedded programs into two phases. The first, platform-independent compiler phase generates E code (code executed by the Embedded Machine), which supervises the timing –not the scheduling-- of application tasks relative to external events,such as clock ticks and sensor interrupts. E code is portable and exhibits, given an input behavior, predictable (i.e., deterministic) timing and output behavior. The second, platform-dependent compiler phase checks the time safety of the E code, that is, whether platform performance (determined by the hardware) and platform utilization (determined by the scheduler of the operating system) enable its timely execution. We have used the Embedded Machine to compile and execute high-performance control applications written in Giotto, such as the flight control system of an autonomous model helicopter. For the first time we flew formation flight for 9 UAVs (7 virtual and 2 real UAV) as demonstrated at the May 03 PI meeting.

Runtime Platform for Timed Multitasking
We checked in a preliminary runtime system for the time multitasking (TM) models in Ptolemy II. The runtime system is mainly a scheduler and some time management; it is written in C and uses POSIX threads. It has been tested under both Cygwin and Redhat Linux. Jie Liu (now at Parc) implemented a helicopter control example using this runtime system and hand generated actor code. The example uses the hardware-in-the-loop simulator that Judy Liebman, Cedric Ma, and Ben Horowitz used to demonstrate Giotto. Jie took the same helicopter, sensor, and actuator code from them, and replaced the Giotto controller with TM

controller. The TM model effectively reduced the one-sample delay per actor introduced by Giotto.

Nonlinear Model Predictive Planning and Control

H. Jin Kim presented nonlinear model predictive planning & control framework which combines trajectory planning and control into a single problem, using ideas from potential-field based navigation for real-time path planning and nonlinear model predictive control for optimal control of nonlinear multi-input, multi-output systems with input/state constraints. She incorporates a tracking performance, potential function, state constraints into the cost function to minimize, and use gradient-descent for on-line optimization. A scenario with five UAVs which are given straight line trajectories that will lead to collision is considered. By using the framework, the safe trajectory of the vehicles are dynamically replanned and tracked by the vehicles under input saturation and state constraints.

We used this work to demonstrate several important experiments:

1. UAV chicken with 2 UAVs flying at each other (hardware experiment demonstrated at PI Meeting May 03)
2. Conflict Resolution for multiple UAVs (software demo showed in May 03) and also to be used in the Softwalls Project.
3. Dogfights with multiple close in UAVs in complex urban environments (software demo given in May 03).

Additionally continuing in the style of the reports of the previous years our detailed accomplishments were as follows:

Time-safety checking for embedded programs

Giotto is a platform-independent language for specifying software for high-performance control applications. We developed a new approach to the compilation of Giotto. Following this approach, the Giotto compiler generates code for a virtual machine, called the E machine, which can be ported to different platforms. The Giotto compiler also checks if the generated E code is time safe for a given platform, that is, if the platform offers sufficient performance to ensure that the E code is executed in a timely fashion that conforms with the Giotto semantics. Time-safety checking requires a schedulability analysis. We showed that while for arbitrary E code, the analysis is exponential, for E code generated from typical Giotto programs, the analysis is polynomial. This supports our claim that Giotto identifies a useful fragment of embedded programs.

Trading probability for fairness

Behavioral properties of control systems can be formalized as objectives in two-player games. Turn-based games model asynchronous interaction between the players (the system and its environment) by interleaving their moves. Concurrent games model synchronous interaction: the players always move simultaneously. Probabilistic moves can be used to model uncertainty in the environment and failure scenarios for the system. Infinitary winning criteria (control objectives) are considered: Buchi, co-Buchi, and more general parity conditions. A generalization of determinacy for parity games to concurrent parity games demands probabilistic (mixed) strategies: either player 1 has a mixed strategy to win with probability 1 almost-sure winning), or player 2 has a mixed strategy to win with positive probability.

We found efficient reductions of concurrent probabilistic Buchi and co-Buchi games to turn-based games with Buchi condition and parity winning condition with three priorities, respectively. From a theoretical point of view, the latter reduction shows that one can trade the probabilistic nature of almost-sure winning for a more general parity (fairness) condition. The reductions improve understanding of concurrent games and provide an alternative simple proof of determinacy of concurrent Buchi and co-Buchi games. From a practical point of view, the reductions turn solvers of turn-based games into solvers of concurrent probabilistic games. Thus improvements in the well-studied algorithms for the former carry over immediately to the latter. In particular, a recent improvement in the complexity of solving turn-based parity games yields an improvement in time complexity of solving concurrent probabilistic co-Buchi games from cubic to quadratic.

The Element of Surprise in Timed Games

The control of real-time and hybrid systems is naturally formulated and solved as timed games (controller versus plant). We developed a symmetric formulation of such timed games and provided the corresponding control algorithms.

We considered concurrent two-person games played in real time, in which the players decide both which action to play, and when to play it.
Such timed games differ from untimed games in two essential ways.
First, players can take each other by surprise, because actions are played with delays that cannot be anticipated by the opponent.
Second, a player should not be able to win the game by preventing time from diverging. We presented a model of timed games that preserves the element of surprise and accounts for time divergence in a way that treats both players symmetrically and applies to all omega-regular winning conditions.

We proved that the ability to take each other by surprise adds extra power to the players. For the case that the games are specified in the style of timed automata, we provided symbolic algorithms for their solution with respect to all omega-regular winning conditions. We also showed that for these timed games, memory strategies are more powerful than memoryless strategies already in the case of reachability objectives.

Schedule Carrying Code

To guarantee the correct execution of a hard real-time program on a given platform, the scheduler must ensure that all deadlines are met.
We introduced the paradigm of schedule-carrying code (SCC), where the compiler proves the existence of a feasible schedule by generating

such a schedule, which is then attached to the generated code in the form of executable instructions that remove the need for a system scheduler. In this way, the schedule is produced once, and revalidated and executed with each use. We evaluated SCC both in theory and practice. In theory, we gave two scenarios, of nonpreemptive and distributed scheduling for Giotto programs, where the generation of a feasible schedule is hard, while the validation of scheduling instructions that are attached to the code is easy. In practice, we implemented SCC and show that explicit scheduling instructions can reduce the scheduling overhead up to 35 percent, and can provide an efficient, flexible, and verifiable means for compiling Giotto on complex architectures, such as the TTA.

Causality in Mixed-Signal and Hybrid Systems

Together with Jie Liu of PARC, we have completed a study of the semantics of causality in mixed-signal and hybrid models [6]. This study extends the application of the Cantor metric as a mathematical tool for defining causality from purely discrete models to mixed-signal and hybrid models. Using the Cantor metric, which represents timed signals, continuous or discrete, in a metric space, we define causality as contractive properties of processes operating on these signals.

Thus, the Banach fixed point theorem can be applied to establish conditions for the existence, uniqueness, and liveness of the behaviors for mixed-signal and hybrid systems. The results also provide theoretical foundations for the simulation technologies for such systems, including the time-marching strategy, evaluation of feedback loops, and the necessity of supporting rollback.

A Giotto-based helicopter control system

We demonstrated the feasibility and benefits of Giotto-based control software development by reimplementing the autopilot system of an autonomously flying model helicopter. Giotto offers a clean separation between the platform independent concerns of software functionality and I/O timing, and the platform dependent concerns of software scheduling and execution. Functionality code such as code computing control laws can be generated automatically from Simulink models or, as in the case of this project, inherited from a legacy system. I/O timing code is generated automatically from Giotto models that specify real-time requirements such as task frequencies and actuator update rates. We extend Simulink to support the design of Giotto models, and from these models, the automatic generation of Giotto code that supervises the interaction of the functionality code with the physical environment. The Giotto compiler performs a schedulability analysis on the Giotto code, and generates timing code for the helicopter platform. The Giotto methodology guarantees the stringent hard real-time requirements of the autopilot system, and at the same time supports the automation of the software development process in a way that produces a transparent software architecture with predictable behavior and reusable components.

Caltech Platform

We continue to work with Caltech on their rolling platform driven by two ducted fans. In particular, Steve Neuendorfer continues to develop code generation with the objective of providing a prototyping software environment for the platform. Caltech has debugged the PIC code for the platform, and has provided us with an instance of the board, but there has not been time yet to do much with it.

A Ptolemy II model of 2D multi-modal helicopter example

John Koo contributed a 2-D multi-modal helicopter example to the Ptolemy II demo library. The model is based on T. J. Koo, S. Sastry "Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization," In Proceedings of IEEE Conference on Decision and Control, Florida, December 1998. The control sequence is derived based on T. J. Koo, G. J. Pappas, and S. Sastry, "Mode Switching Synthesis for Reachability Specifications," Hybrid Systems: Computation and Control, M. D. Di Benedetto and A. Sangiovanni-Vincentelli (Eds.), Lecture Notes in Computer Science, Vol. 2034, pp. 333-346, Springer-Verlag, 2001.

Mobile Models and the Caltech Vehicle

Steve Neuendorffer has demonstrated the use of Ptolemy II for designing controllers for the Caltech ducted fan vehicle. Four configurations have been shown:

(1) the controller and vehicle are both simulated;
(2) the controller is code-generated and the generated code interacts with a simulation of the vehicle (hardware in the loop simulation);
(3) the controller is simulated and interacts with the physical vehicle (simulation-based rapid prototyping); and
(4) the controller is code-generated and the generated code interacts with the physical vehicle (deployment).

To facilitate experimentation with control algorithms, Yang Zhao developed a MobileModel Ptolemy II actor, which is a higher-order actor that accepts a model definition at one of its input ports, and then executes that model to process data provided at the other input ports. The prototype application of this concept uses a Ptolemy II model on the vehicle with the MobileModel actor implementing the control laws, and a separate supervisory model on another laptop providing the control laws over a CORBA channel (using a wireless network).

A key issue for such mobile models is security. Currently, the prototype accepts models provided in cleartext XML and makes no distinction between models that use actors that convey authority (e.g. actors that read or write local files) and models with no such actors.

An essential next step is to develop a security policy for mobile models that enables secure (encrypted) transport of digitally signed executable models and regulated granting of authority to such models.

Hierarchical Hybrid Systems Models

With support from the Mobies program and from NSF, the Ptolemy group at Berkeley developed a "hybrid systems visual modeler" called HyVisual. HyVisual is based on Ptolemy II. As part of the SEC effort, HyVisual has been used to construct various hybrid systems models, and a key semantic issue has been identified (by John Koo).

HyVisual starts with Simulink-like block-diagrams that represent ordinary differential equations. Unlike Simulink, it does not attempt to solve algebraic equations, but rather requires that any directed loop in the model include at least one integrator. This is sufficient to ensure that the execution trace is uniquely defined, at least denotationally (that is, that the denotational semantics is determinate, see [6]). HyVisual augments ODEs with modal models, where a finite-state machine governs discrete switching between modes, and each mode represents a subsystem (called the "refinement" of the mode). The refinement can be a continuous-time model, a discrete model, or a memoryless computation; the information hiding software architecture of Ptolemy II ensures consistent behavior across such heterogeneous models. However, this same information hiding prevents the solver at one level of the hierarchy from knowing the causality properties of another level of the hierarchy.

We have begun working on an abstract representation of causality properties that can be exposed as an interface definition across levels of the hierarchy, and that can be composed so that in a modal model with multiple refinements, the interfaces of the refinements can be merged to define an interface of the modal model. This method will apply not only to the continuous-time ODE-based modeling of HyVisual, but also to discrete-event modeling, synchronous/reactive modeling, and constraint-based models of computation.

More information about HyVisual can be found at:

http://ptolemy.eecs.berkeley.edu/hyvisual/

Localization Technology

David Lee and Paul Yang completed their Ptolemy II implementation of a video-based localization technique for the Caltech ducted fan vehicle. This localization program uses a webcam and JMF (the Java Media Framework). It uses color segmentation to track the vehicle. Two separate color dots are placed on the cart, with one color used for position and a different color for rotation.

During this reporting period, David and Paul implemented:
1) Automatic Color Calibration using a histogram.
2) Distance calculations based on pixel information.

Platform-Based Embedded Software Design and System Integration for Autonomous Vehicles

Automatic control systems typically incorporate legacy code and components that were originally designed to operate independently. Furthermore, they operate under stringent safety and timing constraints. Current design strategies deal with these requirements and characteristics with ad hoc approaches. In particular, when designing control laws, implementation constraints are often ignored or cursorily estimated. Indeed, costly redesigns are needed after a prototype of the control system is built due to missed timing constraints and subtle transient errors. T. J. Koo, J. Liebman, C. Ma, B. Horowitz, A. Sangiovanni-Vincentelli, and S. Sastry used the concepts of platform-based design to develop a methodology for the design of automatic control systems that built in modularity and correct-by-construction procedures. We illustrated our strategy by describing the (successful) application of the methodology to the design of a time-based control system for a helicopter-based Uninhabited Aerial Vehicle (UAV) so the the following objectives can be met:
a. The use of platform-based design allows us to build a bridge between the time-based controller application and the non-time-based sensors and actuators.
b. A time-based controller eliminates the timing irregularities present in first generation system. Further, the Giotto compiler ensures that the controller application meets its timing requirements.
c. Our platform-based design achieves a high degree of modularity. For example, to substitute a different sensor suite in our first redesign requires only changes to the data processor and the data formatting library. The data processor would require a different sensor initialization routine and a new circular buffer; the formatting library would need a new format conversion routine. However, no part of the controller application would need to be changed.

Optimization-based Formation Reconfiguration Planning For Autonomous Vehicles

Given a group of autonomous vehicles, an initial configuration, a final configuration, a set of inter- and intra- vehicle constraints, and a time for reconfiguration, the Formation Reconfiguration Planning problem is focused on determining a nominal input trajectory for each vehicle such that the group can start from the initial configuration and reach its final configuration at the specified time while satisfying the set of inter- and intra- vehicle constraints. S. Zelinski, T. J. Koo, and S. Sastry were interested in solving the Formation Reconfiguration Planning problem for a specific class of systems and a particular form of input signals so that the problem can be reformulated as an optimization problem which can be solved more efficiently, especially for a large group of vehicles. Optimization has proved to be a successful solution to the FRP problem. Our method of implementation is general and portable allowing for use in a wide range of applications for coordinated robots. For example, our method could easily be transported to two dimensions for ground robot coordination. This centralized control scheme has limitations in applications where formations are very large or communication is disrupted. For such applications, a decentralized control scheme is preferred. We are currently working on a decentralized approach to the FRP problem where each vehicle produces its own localized solution based on only local sensor information about its neighboring vehicles. As expected, this is proving to be a more complex problem. Therefore, centralized control is preferred in applications for smaller fully connected formations.

Formation Reconfiguration For Autonomous Vehicles

Coordination of multiple unmanned aerial vehicles (UAVs) poses significant theoretical and technical challenges. Recent advances in sensing, communication and computation enable the conduct of cooperative multiple-UAV mission deemed impossible in the recent past. T. John Koo, Shannon Zelinski and Shankar Sastry worked on solving the Formation Reconfiguration Planning (FRP) problem which is focused on determining a nominal state and input trajectory for each vehicle such that the group can start from the given initial configuration and reach its given final configuration at the specified time while satisfying a set of given inter- and intra- vehicle constraints. Each solution of a FRP problem represents a distinct reconfiguration mode. When coupled with formation keeping modes, they can form a hybrid automaton of formation maneuvers in which a transition from one formation maneuver to another formation maneuver is governed by a finite automaton. We show a simulation results of a group of 9 UAVs performing a sequence of formation reconfigurations.

Collision Avoidance based on Nonlinear Model Predictive Control

H. Jin Kim presented nonlinear model predictive planning and control framework which combines trajectory planning and control into a single problem, using ideas from potential-field based navigation for real-time path planning and nonlinear model predictive control for optimal control of nonlinear multi-input, multi-output systems with input/state constraints. She incorporates a tracking performance, potential function, state constraints into the cost function to minimize, and use gradient-descent for on-line optimization. A scenario with three UAVs which are given straight line trajectories that will lead to collision is considered. By using the framework, the safe trajectory of the vehicles are dynamically replanned and tracked by the vehicles under input saturation and state constraints.

Platform-Based Embedded Software Development
B. Horowitz used the concepts of platform-based design to develop a methodology for the design of automatic control systems that built in modularity and correct-by-construction procedures. The use of platform-based design allows us to build a bridge between the time-based controller application and the non-time-based sensors and actuators. A time-based controller eliminates the timing irregularities present in first generation system. Further, the Giotto compiler ensures that the controller application meets its timing requirements. Our platform-based design achieves a high degree of modularity.

Software Releases and Technology Transfer

- A Simulink model of a helicopter control system, including vehicle dynamics, controllers, sensor models, and Kalman filter, was distributed to Northrop Grumann for the development of High Confidence Control Design for UAVs.
- Code for vision based landing on UAVs for possible use in a FireScout landing demo was also distributed to Northrop Grumann.
- We released Ptolemy II 2.0.1 and Ptplot 5.2. For details, see: http://ptolemy.eecs.berkeley.edu
- Versions of Giotto and the E Machine were released. For Details see http://sec.eecs.berkeley.edu
- Conflict Resolution methods for groups of UAVs and unmanned vehicles are being considered under the Softwall project for integration by Honeywell, Inc. and Boeing in their commercial aircraft companies.
- We helped Boeing script a new scenario for use in the Final Capstone Demo.

Chapter Six: Accomplishments in 2004

There was a fair amount that we accomplished along the three lines of research on the project: formal modeling and verification; embedded software design; and controller design and analysis. We will discuss these here. We are also interested in participating on the Boeing team in the Final Capstone Demo using our vision based and model predictive controller based scenario for collaborative and non-collaborative operations of UAVs.

Formal Modeling and Verification

We performed the following tasks in 03-04:

- Implement and evaluate E (embedded) and S (scheduling) machine on various processors without real-time operating system as intermediary (December 2003-June 2004)

- Develop and implement xGiotto (Extended Giotto) which permits reaction to asynchronous events in addition to time triggered communication (March 2004)

- Formulate a discounted version of temporal logics and solve the corresponding model checking problems to provide algorithms for verifying robust probabilistic hybrid systems. (June 2004).

Embedded Software Design

- Implement the embedded machine on KURT Linux, and adapt KURT Linux to provide us with an extensible and adaptable RTOS infrastructure. (Feb 2004)

- Retarget Ptolemy II code generation to produce code that runs under KURT Linux, including code generated from Giotto, with the FSM domain providing the mode changes and SDF providing the task definitions. (November 2003)

- Develop an interface theory for causality properties and apply it to facilitate hierarchical hybrid system modeling. (May 2004)

- Develop a security policy for mobile models that enables secure (encrypted) transport of digitally signed executable models and regulated granting of authority to such models. (May 2004)

- Develop CORBA and OCP based mobile model mechanisms for distributed control problems. (December 2003).

Control Systems Design

- Implementation of platform based controller for formation flying and formation change maneuvers using a combination of time triggered and asynchronous controllers (August 2003)
- Non-cooperative operations of UAVs using provably correct model predictive algorithms (November 2003)
- Navigation in cluttered urban environments without being stuck in the local minima of potential field based algorithms. (February 2004)

Discounting the Future in Systems Theory

Discounting the future means that the value, today, of a unit payoff is 1 if the payoff occurs today, a if it occurs tomorrow, a^2 if it occurs the day after tomorrow, and so on, for some real-valued discount factor 0 < a < 1. Discounting (or inflation) is a key paradigm in economics and has been studied in Markov decision processes as well as game theory. We submit that discounting also has a natural place in systems engineering: for nonterminating systems, a potential bug in the far-away future is less troubling than a potential bug today. We therefore developed a systems theory with discounting. Our theory includes several basic elements: discounted versions of system properties that correspond to the omega-regular properties, fixpoint-based algorithms for checking discounted properties, and a quantitative notion of bisimilarity for capturing the difference between two states with respect to discounted properties. We developed the theory in a general form that applies to probabilistic systems as well as multicomponent systems (games), but it readily specializes to classical transition systems. We showed that discounting, besides its natural practical appeal, has also several mathematical benefits. First, the resulting theory is robust, in that small perturbations of a system can cause only small changes in the properties of the system. This is of particular relevance for hybrid systems, where the classical models are not robust. Second, the theory is computational, in that the values of discounted properties, as well as the discounted bisimilarity
distance between states, can be computed to any desired degree of precision.

Counter example-guided Control

A major hurdle in the algorithmic verification and control of system is the need to find suitable abstract models, which omit enough details to overcome the state-explosion problem, but retain enough details to exhibit satisfaction or controllability with respect to the specification. The paradigm of counterexample-guided abstraction refinement suggests a fully automatic way of finding suitable abstract models: one starts with a coarse abstraction, attempts to verify or control the abstract model, and if this attempt fails and the abstract counterexample does not correspond to a concrete counterexample, then one uses the spurious counterexample to guide the refinement of the abstract model. We developed a counterexample-guided refinement algorithm for solving omega-regular control objectives. The main difficulty is that in control, unlike in verification, counterexamples are strategies in a game between system

and controller. In the case the controller has no choices, our scheme subsumes known counterexample-guided refinement algorithms for the verification of omega-regular specifications. Our algorithm is useful in all situations where omega-regular games need to be solved, such as supervisory control, sequential and program synthesis, and modular verification. The algorithm is fully symbolic, and therefore applicable also to
infinite-state systems.

Stack-size Analysis for Interrupt-driven Programs

We solved the problems of determining stack boundedness and the exact maximum stack size for three classes of interrupt-driven programs. Interrupt-driven programs are used in many real-time applications that require responsive interrupt handling. In order to ensure responsiveness, programmers often enable interrupt processing in the body of lower-priority interrupt handlers. In such programs a programming error can allow interrupt handlers to be interrupted in cyclic fashion to lead to an unbounded stack, causing the system to crash. For a restricted class of interrupt-driven programs, we showed that there is a polynomial-time procedure to check stack boundedness, while determining the exact maximum stack size is PSPACE-complete. For a larger class of programs, the two problems are both PSPACE-complete, and for the largest class of programs we considered, the two problems are PSPACE-hard and can be solved in exponential time.

Giotto and the E Machine

Steve Neuendorffer, Marco Sanvido, Christoph Kirsch were able to automatically generate code from a Giotto model in Ptolemy, and target the embedded machine. Task code is generated from SDF models, E machine drivers are generated from the connectivity in the model, and the Giotto compiler is used to generate the E machine code. This heavily leveraged Haiyang Zheng's previous work generating Giotto code from Ptolemy. Essentially they filled in the drivers and task code using code generated using Copernicus. Currently, only single mode systemsare supported, and the Giotto model is assumed to be at the top level. This work was cooperative with Mobies.

Open Control Platform

Steve Neuendorffer and Yang Zhao attended the OCP workshop in January at Caltech. They tested some of the concepts that we have been working on, and came away with a few observations. Steve Neuendorffer tested his "crazyboard" controller models, built in Ptolemy II, on the Caltech hardware. There was one laptop sitting on the Crazyboard and running a very simple model to receive control commands, and there was another laptop that runs the control algorithm remotely and sends commands to the laptop on the Crazyboard. The reason for two laptops is that they wanted to test different control algorithms and modify algorithms conveniently. But this approach has a drawback: the latency is quite big due to the control being over the network.

Yang and Steve have suggested an alternative architecture where the control law is implemented in a migrating model that is transported over the network and executed locally. Yang built an experimental setup where a MobileModel actor accepts a model over the network and executes a model to process streaming inputs. The model can be updated in real time by a supervisory model on the second laptop, but the control loop no longer includes the network, thus greatly improving performance. This work continues, with the objective of building an infrastructure that is easy for control systems designers to use.

Yang has also experimented with using JXTA to discover control services and Crazyboard configurations. JXTA is a discovery infrastructure from Sun Microsystems. A key observation that Yang and Steve made is that there is a need for a modeling and simulation environment for developers designing distributed systems based on OCP. Designers today can use Simulink to draw the component diagram of a system, but they cannot do the simulation in Simulink since it does not provide the proper semantics. It would be possible to build a Ptolemy II domain with OCP semantics that might serve this purpose.

Nonlinear Model Predictive Planning and Control

H. Jin Kim presented nonlinear model predictive planning & control framework which combines trajectory planning and control into a single problem, using ideas from potential-field based navigation for real-time path planning and nonlinear model predictive control for optimal control of nonlinear multi-input, multi-output systems with input/state constraints. She incorporates a tracking performance, potential function, state constraints into the cost function to minimize, and use gradient-descent for on-line optimization. A scenario with three UAVs which are given straight line trajectories that will lead to collision is considered. By using the framework, the safe trajectory of the vehicles are dynamically replanned and tracked by the vehicles under input saturation and state constraints.

Optimization-based Formation Reconfiguration Planning For Autonomous Vehicles

Given a group of autonomous vehicles, an initial configuration, a final configuration, a set of inter- and intra-vehicle constraints, and a time for reconfiguration, the Formation Reconfiguration Planning problem is focused on determining a nominal input trajectory for each vehicle such that the group can start from the initial configuration and reach its final configuration at the specified time while satisfying the set of inter- and intra- vehicle constraints. S. Zelinski, T. J. Koo, and S. Sastry were interested in solving the Formation Reconfiguration Planning problem for a specific class of systems and a particular form of input signals so that the problem can be reformulated as an optimization problem which can be solved more efficiently, especially for a large group of vehicles. Optimization has proven to be a successful solution to the FRP problem. Our method of implementation is general and portable allowing for use in a wide range of applications for coordinated robots. For example, our method could easily be transported to two dimensions for ground robot coordination. This centralized control scheme has limitations in applications where formations are very large or communication is disrupted. For such applications, a decentralized control scheme is preferred. We are currently working on a decentralized approach to the FRP problem where each vehicle produces its own localized solution based on only information about its neighboring vehicles. As expected, this is proving to be a more complex problem. Therefore, centralized control is preferred in applications for smaller fully connected formations.

Platform-Based Embedded Software Design and System Integration for Autonomous Vehicles

B. Horowitz, J. Liebman, C. Ma, T. J. Koo, A. Sangiovanni-Vincentelli, and S. Sastry used the concepts of platform-based design to develop a methodology for the design of automatic control systems that built in modularity and correct-by-construction procedures. We illustrated our strategy by describing the application of the methodology to the design of a time-based control system for a helicopter-based Uninhabited Aerial Vehicle (UAV) so that the following objectives can be met:

a. The use of platform-based design allows us to build a bridge between the time-based controller application and the non-time-based sensors and actuators.

b. A time-based controller eliminates the timing irregularities present in first generation system. Further, the Giotto compiler ensures that the controller application meets its timing requirements.

c. Our platform-based design achieves a high degree of modularity. For example, to substitute a different sensor suite in our first redesign requires only changes to the data processor and the data formatting library. The data processor would require a different sensor initialization routine and a new circular buffer; the formatting library would need a new format conversion routine. However, no part of the controller application would need to be changed.

Chapter Seven: Experimentation

Berkeley Experimental Activities

We have a rich track record of performing successful hardware demonstrations on our Berkeley Aerobotics (BEAR) platforms. On the SEC project:we showed

1. Vision based landing (available for transition partners)
2. Learning techniques for complex maneuvers in real time
3. Giotto based embedded software design (available for transition partners)
4. UAV Formation Flight in String Stable configurations available for use in transition role for night flight of manned helicopters such as the Apaches in addition to UAV operations.
5. UAV Chicken for Conflict Detection and Resolution

As we have shown we have now developed some very light weight fixed wing UAVs (the BAT) and some electric rotorcraft for use in cooperative detection of electromagnetic radiation or for cooperative flight path planning in a cluttered environment. We had hardware demonstrations on these between December 2003 and July 2004 as per the following schedule:

1. Electric rotorcraft flight in October 2003
2. Navigation of electric rotorcraft UAV in urban environment including landing December 2003.
3. Navigation and cooperative and non cooperative operations of rotorcraft UAVs in cluttered urban environments May 2004

Berkeley Experiments on the Boeing Open Experimental Platform July 2004:

Air to Air Offensive and Defensive Operations for UAVs

The aim of this experiment was to show some innovative needs for UAVs and UCAVs in future operations using newly developed SEC technology. With the use of multiple UAVs in operations (at last count 22 different types of coalition UAVs were used in Operation Iraqi Freedom OIF), it is important to think about capabilities that are needed to perform UAV missions typically referred to as D^3 (dirty, dull, dangerous). While we have been content thus far with Predators and Global Hawk's being unarmed, and frequently at the mercy of adversarial air forces (especially in a pre-combat phase), it is important to enhance the survivability of UAVs by endowing them with the ability to defend themselves. This is perhaps less than asking for UAVs to perform Air to Air combat operations but requires thought about rapid maneuvers and the ability to evade and engage enemy aircraft which will frequently have a speed advantage and possibly maneuverability and G-force limitations. We feel the need for this capability for both strategic and tactical assets, but our capstone demo experiment will have a tactical conops in mind.

On SEC we have developed numerous techniques for model predictive control of aircraft. A key drawback of model predictive methods thus far has been their inability to compute and refresh with a high refresh rate in close to real time. We have however addressed techniques for rapid re-planning of model predictive receding horizon schemes.

In the three demo experiments that we performed we highlighted both some new technologies and some new capabilities for UAVs.

Experiment 1: UAV as Wingman

The aim of this mission was to have the UAV follow the manually flown F-15E as it performs a sequence of maneuvers:

- Straight and Level Flight
- Steady Climb from one altitude to another
- An acrobatic maneuver such as a barrel roll or a cobra like maneuver.

In this experiment we used the T-33 as the UAV. The F-15E was the manned leader. Its position and velocity vector was communicated to the UAV flight controller as a simulated sensor reading by the UAV of the F-15E position and intent. The F-15E initially started at a fixed altitude 20,000 feet in the middle of the test rangebfollow a straight and level path followed by a climb to 25,000 feet. At the end of the climb the F-15E turned around and perform several repetitions of an acrobatic maneuver and return to the starting point. The UAV started at an altitude of 25,000 feet at a distance of 3 nautical miles from the F-15E at the initial time. It's initial goal will be to close up to the F-15E with an offset of 1 nautical mile in the straight and level path. The tracking performance of the UAV controller was evaluated with respect to a 1 n mile separation requirement.

Success Metrics used were:

- Time delay in forming up (entering formation)
- Tracking accuracy in the climb maneuver
- Success metric to be defined for the acrobatic maneuver including spacing, time delay and overall shape of the UAV maneuver.

Experiment 2: UAV as Evader

The UAV is given a speed disadvantage of 0.8 and maneuverability advantage of more allowable Gs by constraining the capabilities of the Pursuer F-15E. The UAV is asked to confine its maneuvers in a certain envelop of the test range and is given the task of keeping the Pursuer F-15E off its tail. Keeping the F-15E off its tail is defined as keeping the F-15 out of a cone shaped region behind the UAV with the height of the cone being5 nautical miles.

Several evasion strategies were attempted for different pursuer strategies:

- Min-max

- Stackelberg strategy based on a model of the pursuer’s intent.

The UAV and F-15E entered the airspace at different altitudes and at opposite ends of the test range. The goal of the UAV is to traverse the test range diagonally from its entry point to its exit point without being targeted by the F-15E or to stay alive for 10 minutes.
The first experiment called for no prior information about the F-15E pilot (the min-max strategy), the second experiment will call for a prior model of a search strategy being adopted by the pursuer (for example to traverse the test range according to a sweeping pattern), the Stackelberg strategy.

The success metric here was simple: to win the game. This involves either staying alive for 10 minutes or to traverse the test region from corner to corner at a fixed altitude. The reason we have this either/or success metric is to prevent the ability of the F-15E to exclusively guard the “exit” point of the UAV from the test range.

Experiment 3: UAV as a Pursuer

The roles are reversed from Experiment 2 but the speed and maneuverability advantage remain the same and stay with the manned aircraft. Once again the game is won by the evader if it successfully traverses the test range or stays alive.

Conclusion

The overall experiments on the Boeing Open Control Platform were a huge success. The Boeing test pilot frequently lost to our UAV and was openly complimentary of the performance of our combat maneuvers.

Appendix: LIST OF PUBLICATIONS

(by year and alphabetical by first author in each year)

2005
Shim, D., Chung, H., Kim, H.J. and Sastry, S. Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles. *Accepted to AIAA GN&C Conference, San Francisco, August 2005.*

2004
Brooks, C.H., Cataldo, A., Lee, E.A., Liu, J., Liu, X., Neuendorffer, S. and Zheng, H. Eds. HyVisual: A Hybrid System Visual Modeler. *Technical Memorandum UCB/ERL M04/18*, June 28, 2004, University of California, Berkeley, CA 94720.

Brooks, C.H., Lee, E.A., Liu, X., Neuendorffer, S., Zhao, Y. and Zheng, H. Eds. Ptolemy II, Heterogeneous Concurrent Modeling and Design in Java (Volume 1: Introduction to Ptolemy II). *Technical Memorandum UCB/ERL M04/27*, University of California, Berkeley, CA USA 94720, July 29, 2004.

Brooks, C.H., Lee, E.A., Liu, X., Neuendorffer, S., Zhao, Y. and Zheng, H. Eds. Ptolemy II, Heterogeneous Concurrent Modeling and Design in Java (Volume 3: Ptolemy II Domains). *Technical Memorandum UCB/ERL M04/17*, University of California, Berkeley, CA USA 94720, June 24, 2004.

Brooks, C.H., Lee, E.A., Liu, X., Neuendorffer, S., Zhao,Y. and Zheng H. Eds. Ptolemy II, Heterogeneous Concurrent Modeling and Design in Java (Volume 2: Ptolemy II Software Architecture). *Technical Memorandum UCB/ERL M04/16*, University of California, Berkeley, CA USA 94720, June 24, 2004.

Eklund, J.M., Mobassery, F. and Hashutrudi-Zaad, K. Hand Force Estimation Using Fast Orthogonal Search. *Submitted to the 26th Annual Conference IEEE Engineering in Medicine and Biology Society (Sep. 2004)-EMBS 2004.*

Lee, E.A. and Neuendorffer, S. Actor-Oriented Models for Codesign. In Shukla, S. and Talpin, J.P. Eds. *Formal Methods and Models for System Design*, Kluwer, 2004.

Lee, E.A. and Neuendorffer, S. Classes and Subclasses in Actor-Oriented Design. Invited paper, *Conference on Formal Methods and Models for Codesign* (MEMOCODE), June 22-25, 2004, San Diego, CA, USA.

Lee, E.A. and Neuendorffer, S. Concurrent Models of Computation for Embedded Software. *To appear, IEEE Proc. Computers and Digital Techniques, Manuscript date: November 11, 2004*. Previously appeared as *Technical Memorandum UCB/ERL M04/26*, July 22, 2004, University of California, Berkeley, CA 94720.

Lee, E.A. and Xiong, Y. A Behavioral Type System and Its Application in Ptolemy II. *Formal Aspects of Computing 16(3):210-237, 2004. Springer-Verlag, UK.*

Liu, J., Eker, J., Janneck, J.W., Xiaojun, L. and Lee, E.A. Actor-Oriented Control System Design: A Responsible Framework Perspective. *IEEE Transactions on Control Systems Technology 12(2): 250-262, 2004.*

Meingast, M., Geyer, C. and Sastry, S. Vision Based Terrain Recovery for Landing Unmanned Aerial Vehicles. *IEEE Conference on Decision and Control. (Under Review), Dec. 2004.*

Sprinkle, J. and Karsai, G. A Domain-Specific Visual Language for Domain Model Evolution. *J. Vis. Lang. And Comp. 15 (2), 2004.*

Sprinkle, J. Generative Components for Hybrid Systems Tools. *Generative Programming and Component Engineering (GPCE) 2004, (Under Review), Vancouver, BC, Oct. 2004.*

Sprinkle, J. Improving CBS Tool Development with Technological Spaces. *Eleventh IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, Brno, Czech Rep., May 25-27, 2004 (In Press).*

Sprinkle, J. Model-Integrated Computing. *IEEE Potentials 23 (1): 28-30, 2004.*

Sprinkle, J., Eklund, J,M. and Sastry, S. Toward Design Parameterization Support for Model Predictive Control. *Submitted to Autonomous Computing in the Intelligent Systems Design & Applications Conference (Aug. 2004)-ISDA 2004.*

Sprinkle, J., Eklund, J.M., Kim, H.J. and Sastry, S. Encoding Aerial Pursuit/Evasion Games with Fixed Wing Aircraft into a Nonlinear Model Predictive Tracking Controller. *IEEE Conference on Decision and Control 3: 2609-14, 2004.*

Sprinkle, J., Shakernia, O., Miller, R. and Satry, S. Using the Hybrid Systems Interchange Format to Input Design Models to Verification & Validation Tools. *AIAA Conference on Guidance Navigation and Control, Workshop on Validation & Verification, Aug. 2004. Accepted.*

<u>2003</u>

Brooks, C.H. and Lee, E.A. Ptolemy II Coding Style. *Technical Memorandum UCB/ERL M03/44*, University of California at Berkeley, November 24, 2003.

Brooks, C.H., Lee, E.A., Liu, J., Liu, X., Neuendorffer, S., Xiong, Y., Zhao, Y. and Zheng, H. Overview of the Ptolemy Project. *Technical Memorandum UCB/ERL M03/25*, University of California, Berkeley, CA USA 94720, July 2, 2003.

Chatterjee, K., Ma, D., Majumdar R., Zhao, T., Henzinger, T.A. and Palsberg, J. Stack Size Analysis for Interrupt-Driven Programs. *Proceedings of the Tenth International Static Analysis Symposium (SAS). Lecture Notes in Computer Science, 2694, Springer-Verlag: 109-126, 2003.*

Cheong, E., Liebman, J., Liu, J. and Zhao, F. TinyGALS: A Programming Model for Event-Driven Embedded Systems. *Proceedings of the 18th Annual ACM Symposium on Applied Computing (SAC'03)*, Melbourne, FL, Mar. 9-12: pp. 698-704, 2003.

de Alfaro, L., Faella, M., Henzinger, T.A., Majumdar, R. and Stoelinga, M. The Element of Surprise in Timed Games. *Proceedings of the 14th International Conference on Concurrency Theory (CONCUR 03)*. *Lecture Notes in Computer Science 2761, Springer-Verlag: 144-158, 2003.*

de Alfaro, L., Henzinger, T.A. and Majumdar, R. Discounting the Future in Systems Theory. *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP 03)*. *Lecture Notes in Computer Science 2719, Springer-Verlag: 1022-1037, 2003.*

Edwards, S.A., Lee, E.A. The Semantics and Execution of a Synchronous Block-Diagram Language. *Science of Computer Programming 48(1): 21-42, 2003.*

Eker, J., Janneck, J.W., Lee, E.A., Liu, J., Liu, X., Ludvig, J., Neuendorffer, S., Sachs, S. and Xiong, Y. Taming Heterogeneity - the Ptolemy Approach. *Proceedings of the IEEE 91(1): 127-144, 2003.*

Eker. J. and Janneck, J.W. CAL Language Report: Specification of the CAL Actor Language. *Technical Memorandum No. UCB/ERL M03/48*, December 1, 2003, University of California, Berkeley, CA, 94720, USA.

Geyer, C. and Daniilidis, K. Mirrors in Motion: Epipolar Geometry and Motion Estimation. *Proceedings IEEE International Conference on Computer Vision, 2003, Nice France.*

Henzinger, T.A., Horowitz, B. and Kirsch, C.M. Embedded Control Systems Development with Giotto (extended version). *Proceedings of the ACM International Workshop on Languages, Compilers, and Tools for Embedded Systems. LCTES:64-72, 2001. In “Software-Enabled Control: Information Technology for Dynamical Systems”, Samad, T. and Balas, G. Eds. IEEE press and Wiley Interscience: 123-146, 2003.*

Henzinger, T.A., Horowitz, B. and Kirsch, C.M. Giotto: A Time-Triggered Language for Embedded Programming. *Proceedings of the IEEE 91:84-99, 2003.*

Henzinger, T.A., Jhala, R. and Majumdar, R. Counterexample-Guided Control. *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming (ICALP)*. *Lecture Notes in Computer Science, 2719, Springer-Verlag: 886-902, 2003.*

Henzinger, T.A., Kirsch, C.M., Sanvido, M.A.A. and Pree. W. From Control Models to Real-Time Code Using Giotto. *IEEE Control Systems Magazine 23(1): 50-64, 2003.*

Horowitz, B., Liebman, J., Ma, C., Koo, T.J., Sangiovanni-Vincentelli, A. and Sastry S. Platform-Based Embedded Software Design and System Integration for Autonomous Vehicles. *Proceedings of the IEEE 91(1): 198-211, 2003.*

Karsai, G., Agrawal, A., Shi, F. and Sprinkle, J. On the Use of Graph Transformations in the Formal Specification of Model Interpreters. *Journal of Universal Computer Science 9 (11): 1296-1321, 2003.*

Lee, E.A. and Xiong, Y. A Behavioral Type System and Its Application in Ptolemy II. To appear in *Aspects of Computing Journal, special issue on "Semantic Foundations of Engineering Design Languages." This version: November 10, 2003.*

Lee, E.A., Neuendorffer, S. and Wirthlin, M.J. Actor-Oriented Design of Embedded Hardware and Software Systems. Invited paper, *Journal of Circuits, Systems, and Computers 12 (3): 231-260, 2003.*

Liu, J. and Lee, E.A. On the Causality of Mixed-Signal and Hybrid Models. *Proceedings of the 6th International Workshop on Hybrid Systems: Computation and Control (HSCC '03), April 3-5, 2003, Prague, Czech. Lecture Notes in Computer Science 2623, Springer-Verlag: 328-342, 2003.*

Liu, J. and Lee, E.A. Timed Multitasking for Real-Time Embedded Software. *Special issue on "Advances in Software Enabled Control," IEEE Control Systems Magazine 23 (1): 65-75, 2003.*

Liu, X., Liu, J., Eker, J. and Lee, E.A. Heterogeneous Modeling and Design of Control Systems. *In "Software-Enabled Control: Information Technology for Dynamical Systems", Samad, T. and Balas, G. Eds. Wiley-IEEE Press, April 2003.*

Neuendorffer, S. Implementation Issues in Hybrid Embedded Systems. *Technical Memorandum No. UCB/ERL M03/22*, University of California, Berkeley, CA, 94720, USA, June 24, 2003.

Ng, A., Kim, H.J., Jordan, M. and Sastry, S. Autonomous Helicopter Flight Via Reinforcement Learning. *16th Neural Information Processing Systems, December 2003.*

Nordstrom, S., Shetty, S., Chhokra, K.G., Sprinkle, J., Eames, B. and Ledeczi, A. ANEMIC: Automatic Interface Enabler for Model Integrated Computing. *Lecture Notes in Computer Science 2830: 138-150, 2003.*

Shim, D.H., Kim, H.J. and Sastry, S. Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots. *IEEE Conference on Decision and Control, December 2003.*

Sprinkle, J. and Karsai, G. Model Migration Through Visual Modeling. *OOPSLA, Anaheim, CA, Oct. 26, 2003.*

Sprinkle, J. Managing Intent: Propagation of Meaning During Model Transformations. *UML 2003, San Francisco, CA, Oct. 21, 2003.*

Varma, A. Retargetable Optimizing Java-to-C Compiler for Embedded Systems. *Master's Report, Department of Electrical and Computer Engineering*, University of Maryland, College Park, MD, 2003.

Zelinski, S., Koo, T.J., Sastry, S. 2003 IEEE International Conference on Robotics and Automation. *IEEE 3: 3758-3763, 2003.*

2002

Bhattacharyya, S.S., Cheong, E., Davis II, J., Goel, M., Hylands, C., Kienhuis, B., Lee, E.A., Liu, J., Liu, X., Muliadi, L., Neuendorffer, S., Reekie, J., Smyth, N., Tsay, J., Vogel, B., Williams, W., Xiong, Y. and Zheng, H. Heterogeneous Concurrent Modeling and Design in Java. *Memorandum UCB/ERL M02/23*, University of California, Berkeley, CA USA 94720, August 5, 2002.

Cassez, F., Henzinger, T.A., Raskin, J.F. A Comparison of Control Problems for Timed and Hybrid Systems. *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control 2002 (HSCC 2002)*. *Lecture Notes in Computer Science 2289, Springer-Verlag: 134-148, 2002.*

Henzinger, T.A. and Kirsch, C.M. The Embedded Machine: Predictable, Portable Real-Time Code. *Proceedings of the International Conference on Programming Language Design and Implementation (PLDI 2002). ACM Press: 315-326, 2002.*

Henzinger, T.A., Kirsch, C.M., Majumdar, R. and Matic, S. Time-Safety Checking for Embedded Programs. *Proceedings of the Second International Workshop on Embedded Software (EMSOFT 2002). Lecture Notes in Computer Science 2491, Springer-Verlag: 76-92, 2002.*

Horowitz, B., Liebman, J., Ma, C., Koo, T.J., Henzinger, T.A., Sangiovanni-Vincentelli, A.L. and Sastry, S. Embedded Software Design and System Integration for Rotorcraft UAV Using Platforms. *Proceedings of the 15th IFAC World Congress on Automatic Control. Elsevier Science, 2002.*

Horowitz, B., Liebman, J., Ma, C., Tal, R., Koo, T.J., Henzinger, T.A. and Sastry, S. Hardware-in-the-Loop (HIL) Simulation of Multi-Vehicle Multi-Modal Embedded Systems. *Proc. IFAC World Congress on Automatic Control. Barcelona, Spain, 2002.*

Jurdzinski, M., Kupferman, O. and Henzinger, T.A. Trading Probability for Fairness. *Proceedings of the International Conference for Computer Science Logic (CSL 2002). Lecture Notes in Computer Science 2471, Springer-Verlag: 292-305, 2002.*

Koo, J.T., Liebman, J., Ma, C., Horowitz, B., Sangiovanni-Vincentelli, A. and Sastry, S. Platform-Based Embedded Software Design for Multi-Vehicle Multi-Modal Systems. *Proceedings of the Second International Conference on Embedded Software (EMSOFT 2002). Lecture Notes in Computer Science 2491, Springer-Verlag: 32-45, 2002.*

Koo, T. J, Pappas, G.J. and Sastry, S. Multi-Modal Control of Constrained Nonlinear Systems Software-Enabled Control: Information Technology for Dynamical Systems. *Samad, T. and Balas, G. Eds. IEEE Press, 2002.*

Koo, T.J. and Sastry S. Bisimulation Based Hierarchical System Architecture for Single-Agent Multi-Modal Systems. *Proceedings of the 5th International Workshop, on Hybrid Systems: Computation and Control (HSCC 2002). Lecture Notes in Computer Science 2289, Springer-Verlag: 281-293, 2002.*

Lee, E.A. and Xiong, Y. Behavioral Types for Component-Based Design. *Technical Memorandum UCB/ERL M02/29*, University of California, Berkeley, CA 94720, USA, September 27, 2002.

Liu, J. and Lee, E.A. A Component-Based Approach to Modeling and Simulating Mixed-Signal and Hybrid Systems. *ACM Transactions on Modeling and Computer Simulation. Special issue on Computer Automated Multi-Paradigm Modeling 12 (4): 343-368, 2002.*

Liu, J., Eker, J., Janneck, J.W. and Lee, E.A. Realistic Simulations of Embedded Control Systems. *International Federation of Automatic Control, 15th IFAC World Congress*, *Barcelona, Spain, July 21-26, 2002.*

Murthy, P.K. and Lee, E.A. Multidimensional Synchronous Dataflow. *IEEE Transactions on Signal Processing 50(8):2064 -2079, 2002.*

Schobbens, P.Y., Raskin, J.F., Henzinger, T.A. and Ferier, L. Axioms for Real-Time Logics. *Theoretical Computer Science 274: 151-182, 2002.*

2001

Brown, T., Pasetti, A., Pree, W., Henzinger, T.A. and Kirsch, C.M. A Reusable and Platform-Independent Framework for Distributed Control Systems. *Proceedings of the 20th*

Annual IEEE/AIAA Digital Avionics Systems Conference 2001 (DASC 2001). IEEE Press 2: 1-11, 2001.

Davis II, J., Hylands, C., Kienhuis, B., Lee, E.A., Liu, J., Liu, X., Muliadi, L., Neuendorffer, S., Tsay, J., Vogel, B. and Xiong, Y. Heterogeneous Concurrent Modeling and Design in Java. *Technical Memorandum UCB/ERL M01/12, EECS*, University of California, Berkeley, March 15, 2001.

de Alfaro, L., Henzinger, T.A. and Jhala, R. Compositional Methods for Probabilistic Systems. *Proceedings of the 12th International Conference on Concurrency Theory 2001 (CONCUR 2001). Lecture Notes in Computer Science 2154, Springer-Verlag: 351-365, 2001.*

de Alfaro, L., Henzinger, T.A. and Majumdar, R. From Verification to Control: Dynamic Programs for Omega-Regular Objectives. *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science 2001, Los Alamitos, CA, USA. IEEE Comput. Soc. 2001: 279-290.*

de Alfaro, L., Henzinger, T.A. and Majumdar, R. Symbolic Algorithms for Infinite-State Games. *Proceedings of the 12th International Conference on Concurrency Theory (CONCUR 2001). Lecture Notes in Computer Science 2154, Springer-Verlag: 536-550, 2001. (Best paper award).*

de Alfaro, L., Henzinger, T.A. and Mang, F.Y.C. The Control of Synchronous Systems II. *Proceedings of the 12th International Conference on Concurrency Theory (CONCUR 2001). Lecture Notes in Computer Science 2154, Springer-Verlag: 566-581. 2001.*

Eker J., Fong, C., Janneck, J.W. and Liu, J. Design and Simulation of Heterogeneous Control Systems Using Ptolemy II. *Proceedings of IFAC Conference on New Technologies for Computer Control (NTCC 2001). Univ. Hong Kong: pp. 271-276. Hong Kong, China 2001.*

Henzinger, T.A. and Kirsch, C.M. Eds. Embedded Software. *Proceedings of the First International Workshop EMSOFT 01. Lecture Notes in Computer Science 2211, Springer-Verlag, 2001. ISBN 3-540-42673-6, 504 pages.*

Henzinger, T.A., Minea, M. and Prabhu, V. Assume-Guarantee Reasoning for Hierarchical Hybrid Systems. *Proceedings of the Fourth International Workshop on Hybrid Systems: Computation and Control (HSCC 2001). Lecture Notes in Computer Science 2034, Springer-Verlag: 275-290, 2001.*

Henzinger, T.A., Preussig, J. and Wong-Toi, H. Some Lessons from the HyTech Experience. *Proceedings of the 40th Annual Conference on Decision and Control (CDC 2001). IEEE Press: pp. 2887-2892. December 2001.*

Hu, J., Prandini, M., Johansson, K.H. and Sastry, S. Hybrid Geodesics as Optimal Solutions to the Collision-Free Motion Planning Problem. *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science 2034, Springer-Verlag: 305-318, 2001.*

Koo, T.J, Pappas, G.J. and Sastry, S. Multi-Modal Control of Systems with Constraints. *Proceedings of the 40th IEEE Conference on Decision and Control. IEEE. Part 3: 2075-2080, 2001.*

Koo, T.J. Hierarchical System Architecture for Multi-Agent Multi-Modal Systems. *Proceedings of the 40th IEEE Conference on Decision and Control. IEEE. Part 2: 1509-1514. 2001.*

Koo, T.J., Liebman, J., Ma, C. and Sastry, S. Hierarchical Approach for Design of Multi-Vehicle Multi-Modal Embedded Software. *First Workshop on Embedded Software, EMSOFT 2001. Lecture Notes in Computer Science 2211, Springer-Verlag: 344-360. 2001.*

Lee, E.A. and Xiong, Y. System-Level Types for Component-Based Design. *First Workshop on Embedded Software, EMSOFT2001. Lake Tahoe, CA, USA, Oct. 8-10, 2001.*

Lee, E.A. Embedded Software. *Technical Memorandum UCB/ERL M01/26*, University of California, Berkeley, CA 94720, July 12, 2001. (joint work with Mobies)

Lee, E.A. Soft Walls - Modifying Flight Control Systems to Limit the Flight Space of Commercial Aircraft (*Draft 2*). *Revised from Technical Memorandum UCB/ERL M01/31*, University of California, Berkeley, CA 94720, October 3, 2001.

Liu, J. Responsible Frameworks for Heterogeneous Modeling and Design of Embedded Systems. *Ph.D. thesis, Technical Memorandum UCB/ERL M01/41*, University of California, Berkeley, CA 94720, December 2001.

Liu, J., Jefferson, S. and Lee, E.A. Motivating Hierarchical Run-Time Models in Measurement and Control Systems. *Proceedings of the 2001 American Control Conference*, *Arlington, VA, June 25-27: pp. 3457-3462. 2001.*

Liu, J., Liu, X. and Lee, E.A. Modeling Distributed Hybrid Systems in Ptolemy II. *In the embedded tutorial of the 2001 American Control Conference*, *Arlington, VA, June 25-27. pp. 4984-4985. 2001.*

Liu, X., Xiong, Y. and Lee, E.A. The Ptolemy II Framework for Visual Languages. Poster paper, *Symposium on Visual Languages and Formal Methods, Stresa, Italy, Sept. 5-7, 2001.*

Shakernia O., Pappas, G.J. and Sastry, S. Semi-Decidable Synthesis for Triangular Hybrid Systems. *Proceedings of the 4th International Workshop on Hybrid Systems, Computation and Control (HSCC 2001). Lecture Notes in Computer Science 2034, Springer-Verlag: 487-500, 2001.*

Vidal. R., Schaffert, S., Shakernia, O., Lygeros, J. and Sastry, S. Decidable and Semi-Decidable Controller Synthesis for Classes of Discrete Time Hybrid Systems. *Proceedings of the 40th IEEE Conference on Decision and Control. IEEE. Part 2: 1243-1248. 2001.*

Whitaker, P. The Simulation of Synchronous Reactive Systems In Ptolemy II. Master's Report, *Memorandum UCB/ERL M01/20*, Electronics Research Laboratory, University of California, Berkeley, May 2001.

Zhang, J., Johansson, K.H., Lygeros, J. and Sastry, S**.** Zeno Hybrid Systems**.** *International Journal of Robust & Nonlinear Control 11(5): 435-451, 2001.*

2000

Alur, R., Henzinger, T.A., Lafferriere, G. and Pappas, G.J. Discrete Abstractions of Hybrid Systems. *Proceedings of the IEEE 88: 971-984, 2000.*

Balluchi, A., Benvenuti, L., Di Benedetto, M.D., Pinello, C. and Sangiovanni-Vincentelli, A.L. Automotive Engine Control and Hybrid Systems: Challenges and Opportunities. *Proceedings of the IEEE, Hybrid Systems: Theory and Applications, July 2000.*

Bhattacharyya, S.S., Sriram, S. and Lee, E.A. Resynchronization for Multiprocessor DSP Systems. *IEEE Transactions on Circuit and Systems - I: Fundamental Theory and Applications 47(11), 2000.*

de Alfaro, L., Henzinger, T.A. and Mang, F.Y.C. Detecting Errors Before Reaching Them. *Proceedings of the 12th International Conference on Computer-aided Verification (CAV 00)*. *Lecture Notes in Computer Science 1855, Springer-Verlag: 186-201, 2000.*

de Alfaro, L., Henzinger, T.A. and Mang, F.Y.C. The Control of Synchronous Systems. *Proceedings of the 11th International Conference on Concurrency Theory 2000 (CONCUR 00)*. *Lecture Notes in Computer Science 1877, Springer-Verlag: 458-473, 2000.*

de Alfaro, L., Henzinger, T.A. Concurrent Omega-Regular Games. *Proceedings Fifteenth Annual IEEE Symposium on Logic in Computer Science*, *Los Alamitos, CA, USA. (Cat. No.99CB36332). IEEE Comput. Soc.: pp. 141-154, 2000.*

Egerstedt M., Ogren, P., Shakernia, O. and Lygeros, J. Toward Optimal Control of Switched Linear Systems. *Proceedings of the 39th IEEE Conference on Decision and Control. IEEE. Part 1: 587-592, 2000.*

Henzinger, T.A. and Majumdar, R. A Classification of Symbolic Transition Systems. *Proceedings of the 17th International Conference on Theoretical Aspects of Computer Science (STACS 2000)*. *Lecture Notes in Computer Science 1770, Springer-Verlag: 13-34, 2000.*

Henzinger, T.A. and Majumdar, R. Symbolic Model Checking for Rectangular Hybrid Systems. *Proceedings of the Sixth International Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2000)*. *Lecture Notes in Computer Science 1785, Springer-Verlag: 142-156, 2000.*

Henzinger, T.A. and Raskin, J.F. Robust Undecidability of Timed and Hybrid Systems. *Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control (HSCC 2000)*. *Lecture Notes in Computer Science 1790, Springer-Verlag: 145-159, 2000.*

Henzinger, T.A. Masaccio: A Formal Model for Embedded Components. *Proceedings of the First IFIP International Conference on Theoretical Computer Science (TCS 00)*. *Lecture Notes in Computer Science 1872, Springer-Verlag: 549-563, 2000.*

Henzinger, T.A., Horowitz, B., Majumdar, R. and Wong-Toi, H. Beyond HyTech: Hybrid Systems Analysis Using Interval Numerical Methods. *Proceedings of the Third International Workshop on Hybrid Systems: Computation and Control (HSCC 2000). Lecture Notes in Computer Science 1790, Springer-Verlag: 130-144, 2000.*

Henzinger, T.A., Majumdar, R., Mang, F.Y.C. and Raskin, J.F. Abstract Interpretation of Game Properties. *Proceedings of the Seventh International Static Analysis Symposium (SAS 00). Lecture Notes in Computer Science 1824, Springer-Verlag: 220-239, 2000.*

Koo, T.J. Hybrid System Design and Embedded Controller Synthesis for Multi-Modal Control. *PhD Thesis, EECS Dept.* University of California, Berkeley, August, 2000.

Lee, B. Specification and Design of Reactive Systems. *Ph.D. thesis, Memorandum UCB/ERL M00/29, Electronics Research Laboratory*, University of California, Berkeley, May, 2000.

Liu, J. and Lee, E.A. Component-Based Hierarchical Modeling of Systems with Continuous and Discrete Dynamics. *Proceedings of the 2000 IEEE International Conference on Control Applications and IEEE Symposium on Computer-Aided Control System Design (CCA/CACSD'00), Anchorage, AK, September 25-27: pp. 95-100, 2000.*

Shakernia O., Pappas, G.J. and Sastry, S. Semidecidable Controller Synthesis for Classes of Linear Hybrid Systems. *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*. *IEEE. Part 2: 1834-1839, 2000.*

Shakernia, O., Pappas, G.J. and Sastry, S. Decidable Controller Synthesis for a Class of Linear Systems. *In Hybrid Systems: Computation and Control (HSCC'00), Pittsburgh, PA, March, 2000.*

Simic, S., Johansson, K.H., Sastry, S. and Lygeros, J. Towards a Geometric Theory of Hybrid Systems. *In Hybrid Systems: Computation and Control (HSCC'00), Pittsburgh, PA, March 2000.*

Tsay, J. A Code Generation Framework for Ptolemy II. *ERL Technical Report UCB/ERL No. M00/25, Dept. EECS*, University of California, Berkeley, CA 94720, May 19, 2000.

Tsay, J., Hylands, C. and Lee, E.A. A Code Generation Framework for Java Component-Based Designs. *CASES '00, November 17-19, 2000, San Jose, CA.*

Zhang, J., Johansson, K.H., Lygeros, J. and Sastry, S. Dynamical Systems Revisited: Hybrid Systems with Zeno Executions. *In Hybrid Systems: Computation and Control, Krogh and Lynch, N. Eds. Lecture Notes in Computer Science 1790, Springer-Verlag: 451-464, 2000.*

<u>1999</u>

Bhattacharyya, S.S., Murthy, P.K. and Lee, E.A. Synthesis of Embedded Software from Synchronous Dataflow Specifications," *Journal of VLSI Signal Processing Systems*, *21(2), 1999.*

Egerstedt, M., Johansson, K., Lygeros, J. and Sastry, S. Behavior Based Robotics Using Regularized Hybrid Automata. *Proceedings of the 38th IEEE Conference on Decision and Control*. *IEEE. Part 4: 3400-3405, 1999.*

Girault, A., Lee, B. and Lee, E.A. Hierarchical Finite State Machines with Multiple Concurrency Models. *IEEE Transactions On Computer-aided Design Of Integrated Circuits And Systems 18(6), 1999 (revised from Memorandum UCB/ERL M97/57, Electronics Research Laboratory, University of California, Berkeley, CA 94720, August 1997).*

Johansson, K.H., Lygeros, J., Sastry, S. and Egerstedt, M. Simulation of Zeno Hybrid Automata. *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*. *IEEE. Part 4: 3538-3543, 1999.*

Lee, E.A. Modeling Concurrent Real-Time Processes Using Discrete Events. Invited paper, *Annals of Software Engineering*, *Special Volume on Real-Time Software Engineering 7: 25-45, 1999. Revised from UCB/ERL Memorandum M98/7, March 4th 1998.*

Liu, J., Liu, X., Koo, T.J., Sinopoli, B., Sastry, S. and Lee, E.A. A Hierarchical Hybrid System Model and Its Simulation. *Proceedings of the 38th IEEE Conference on Decision and Control, Dec. 1999, Phoenix, AZ.*

Liu, J., Wu, B., Liu, X. and Lee, E.A. Interoperation of Heterogeneous CAD Tools in Ptolemy II. *Symposium on Design, Test, and Microfabrication of MEMS/MOEMS, March 1999, Paris, France.*

Lygeros J., Johansson, K.H., Sastry, S. and Egerstedt, M. On the Existence of Executions of Hybrid Automata. *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304). IEEE. Part 3: 2249-2254. 1999.*

Najjar, W.A., Lee, E.A. and Gao, G.R. Advances in the Dataflow Computational Model. *Parallel Computing 25: 1907-1929,1999.*

1998

Bhattacharyya, S.S., Sriram, S. and Lee, E.A. Resynchronization for Multiprocessor DSP Implementation - Part 1: Maximum-Throughput Resynchronization. *Tech. Rep., Digital Signal Processing Laboratory, University of Maryland, College Park, July 1998. Revised from Memorandum UCB/ERL 96/55, Electronics Research Laboratory, University of California at Berkeley, October, 1996.*

Bhattacharyya, S.S., Sriram, S. and Lee, E.A. Resynchronization for Multiprocessor DSP Implementation - Part 2: Latency-Constrained Resynchronization. *Tech. Rep., Digital Signal Processing Laboratory, University of Maryland, College Park, July 1998. Revised from Memorandum UCB/ERL 96/56, Electronics Research Laboratory, University of California at Berkeley, October, 1996.*

Lee, B. and Lee, E.A. Hierarchical Concurrent Finite State Machines in Ptolemy. *Proc. of International Conference on Application of Concurrency to System Design*, *p. 34-40, Fukushima, Japan, March 1998.*

Lee, B. and Lee, E.A. Interaction of Finite State Machines with Concurrency Models. *Proc. of Thirty Second Annual Asilomar Conference on Signals, Systems, and Computers*, *Pacific Grove, California, November 1998.*

Lee, E.A. and Sangiovanni-Vincentelli, A. A Framework for Comparing Models of Computation. *IEEE Transactions on CAD 17 (12), 1998. Revised from ERL Memorandum UCB/ERL M97/11, University of California, Berkeley, CA 94720, January 30, 1997).*

Reekie, H.J., Lee, E.A. The Tycho Slate: Complex Drawing and Editing in Tcl/Tk. *Proceedings of the Sixth Annual Tcl/Tk Conference*. *USENIX Assoc. 1998, pp.37-46. Berkeley, CA, USA.*

Smyth, N. Communicating Sequential Processes Domain in Ptolemy II. *Master's Report, UCB/ERL Memorandum M98/70, Dept. of EECS*, University of California, Berkeley, CA 94720, December 1998.